SIEMENS

SINEC

SINEC S1
Masterbaugruppe CP 2413

C79000-G8900-C061

Ausgabe 01
Band 1 von 1

Einführung 1

Masterbaugruppe CP 2413
und AS-I 2413 MS-DOS/WINDOWS 2

Anhang

PICS A

Abkürzungen und Begriffe B

Literaturverzeichnis C

Hinweise zur CE-Kennzeichnung D

SINEC ist ein Warenzeichen von SIEMENS

Siemens Aktiengesellschaft

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so daß wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in der Druckschrift werden jedoch regelmäßig überprüft. Notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten. Für Verbesserungsvorschläge sind wir dankbar.

Technische Änderungen vorbehalten.

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte vorbehalten, insbesondere für den Fall der Patenterteilung oder GM-Eintragung.

We have checked the contents of this manual for agreement with the hardware described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in this manual are reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvement are welcome.

Technical data subject to change.

The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility or design, are reserved.

Nous avons vérifié la conformité du contenu du présent manuel avec le matériel et le logiciel qui y sont décrits. Or, des divergences n'étant pas exclues, nous ne pouvons pas nous porter garants pour la conformité intégrale. Si l'usage du manuel devait révéler des erreurs, nous en tiendrons compte et apporterons les corrections nécessaires dès la prochaine édition. Veuillez nous faire part de vos suggestions.
Nous nous réservons le droit de modifier les caractéristiques techniques.

Toute communication ou reproduction de ce support d'informations, toute exploitation ou communication de son contenu sont interdites, sauf autorisation expresse. Tout manquement à cette règle est illicite et expose son auteur au versement de dommages et intérêts. Tous nos droits sont réservés, notamment pour le cas de la délivrance d'un brevet ou celui de l'enregistrement d'un modèle d'utilité.

Copyright © Siemens AG 1995 All Rights Reserved

Siemens Aktiengesellschaft

Elektronikwerk Karlsruhe

Printed in the Federal Republic of Germany

SINEC

SINEC S1 Masterbaugruppe CP 2413

Gerätehandbuch C79000-B8900-C084

Wichtiger Hinweis

Wir weisen darauf hin, daß der Inhalt dieser Betriebsanleitung nicht Teil einer früheren oder bestehenden Vereinbarung, Zusage oder eines Rechtsverhältnisses ist oder diese abändern soll. Sämtliche Verpflichtungen von Siemens ergeben sich aus dem jeweiligen Kaufvertrag, der auch die vollständige und allein gültige Gewährleistungsregel enthält. Diese vertraglichen Gewährleistungsbestimmungen werden durch die Ausführungen dieser Betriebsanleitung weder erweitert noch beschränkt.

Wir weisen außerdem darauf hin, daß aus Gründen der Übersichtlichkeit in dieser Betriebsanleitung nicht jede nur erdenkliche Problemstellung im Zusammenhang mit dem Einsatz dieses Gerätes beschrieben werden kann. Sollten Sie weitere Informationen benötigen oder sollten besondere Probleme auftreten, die in der Betriebsanleitung nicht ausführlich genug behandelt werden, können Sie die erforderliche Auskunft über die örtliche Siemens-Niederlassung anfordern.

Allgemeines

Dieses Gerät wird mit Elektrizität betrieben. Beim Betrieb elektrischer Geräte stehen zwangsläufig bestimmte Teile dieser Geräte unter gefährlicher Spannung.

WARNUNG !

! Bei Nichtbeachtung der Warnhinweise können deshalb schwere Körperverletzungen und/oder Sachschäden auftreten.

Nur entsprechend qualifiziertes Personal sollte an diesem Gerät oder in dessen Nähe arbeiten. Dieses Personal muß gründlich mit allen Warnungen und Instandhaltungsmaßnahmen gemäß dieser Betriebsanleitung vertraut sein.

Der einwandfreie und sichere Betrieb dieses Gerätes setzt sachgemäßen Transport, fachgerechte Lagerung und Montage sowie sorgfältige Bedienung und Instandhaltung voraus.

Anforderungen an die Qualifikation des Personals

Qualifiziertes Personal im Sinne dieser Betriebsanleitung bzw. der Warnhinweise sind Personen, die mit Aufstellung, Montage, Inbetriebsetzung und Betrieb dieses Produktes vertraut sind und über die ihrer Tätigkeit entsprechenden Qualifikation verfügen, wie z.B.:

- Ausbildung oder Unterweisung bzw. Berechtigung, Stromkreise und Geräte bzw. Systeme gemäß den aktuellen Standards der Sicherheitstechnik ein- und auszuschalten, zu erden und zu kennzeichnen;

- Ausbildung oder Unterweisung gemäß den aktuellen Standards der Sicherheitstechnik in Pflege und Gebrauch angemessener Sicherheitsausrüstungen;

- Schulung in Erster Hilfe.

INHALTSVERZEICHNIS

1 EINFÜHRUNG 1-1

1.1 Allgemeines 1-2
1.1.1 Kapitelübersicht 1-2
1.1.2 Allgemeine Lesehinweise und Schreibregeln 1-2
1.1.3 Voraussetzungen 1-2
1.1.4 Weitere Informationen 1-2
1.1.5 Hotline 1-2

2 MASTERBAUGRUPPE CP 2413 UND AS-I-2413 MS-DOS/WINDOWS 2-1

2.1 Übersicht AS-I-2413 MS-DOS/Windows 2-2

2.2 Ablaufumgebung und Anforderungen 2-3
2.2.1 Hardwareanforderungen 2-3
2.2.2 Softwarevoraussetzungen 2-3

2.3 Übersicht CP 2413 2-4
2.3.1 Hardwareinstallation 2-4
2.3.2 Technische Daten 2-5
2.3.3 Anschluß der AS-I-Leitung 2-6

2.4 Softwareinstallation 2-7
2.4.1 Was Sie zur Installation brauchen 2-7
2.4.2 Wie Sie vorgehen 2-8
2.4.3 Was Sie beachten müssen 2-10
2.4.4 Hantierungsmöglichkeiten nach der Installation 2-11
2.4.4.1 SINEC-Setup 2-11
2.4.4.2 Autorisierung übertragen und prüfen 2-12
2.4.4.3 Konfiguration des CP 2413 2-13
2.4.5 Verzeichnisstruktur 2-15
2.4.6 Initialisieren des CP2413, die Start-Batch-Datei 2-15
2.4.7 AS-I-Treiber 2-15
2.4.8 AS-I-Monitor 2-16
2.4.9 Die AS-I-Library 2-17
2.4.10 Beispielprogramme 2-17

2.5 Die AS-I-Anwenderschnittstelle 2-18
2.5.1 Wirkungsweise der Aufträge 2-18
2.5.2 Funktionsübersicht 2-23
2.5.2.1 asi_init 2-26
2.5.2.2 asi_shut 2-27
2.5.2.3 asi_read_idi (Eingangsdatenabbild_lesen) 2-28
2.5.2.4 asi_write_odi (Ausgangsdatenabbild_schreiben) 2-30
2.5.2.5 asi_set_permanent_parameter (Parameterwert_projektieren) 2-32
2.5.2.6 asi_get_permanent_parameter (Projektierten_Parameterwert_lesen) 2-34
2.5.2.7 asi_write_parameter (Parameterwert_schreiben) 2-36
2.5.2.8 asi_read_parameter (Parameterwert_lesen) 2-38
2.5.2.9 asi_store_actual_parameters (Ist_Parameterwerte_projektieren) 2-40
2.5.2.10 asi_set_permanent_configuration (Konfigurationsdaten_Projektieren) 2-41
2.5.2.11 asi_get_permanent_configuration (Projektierte_Konfigurationsdaten_Lesen) 2-43
2.5.2.12 asi_store_actual_configuration (Ist_Konfigurationsdaten_Projektieren) 2-45
2.5.2.13 asi_read_actual_configuration (Ist_Konfigurationsdaten_Lesen) 2-46
2.5.2.14 asi_set_lps (LPS_Projektieren) 2-48
2.5.2.15 asi_get_lps (LPS_Lesen) 2-49
2.5.2.16 asi_get_las (LAS_Lesen) 2-50
2.5.2.17 asi_get_lds (LDS_Lesen) 2-51

2.5.2.18 asi_get_flags (Flags_Lesen) 2-52
2.5.2.19 asi_set_operation_mode (Projektierungsmodus_Setzen) 2-54
2.5.2.20 asi_set_offline_mode (Offlinemodus_Setzen) 2-55
2.5.2.21 asi_change_slave_address (Betriebsadresse_Ändern) 2-56
2.5.2.22 asi_reset 2-58
2.5.2.23 asi_execute_command 2-59
2.5.2.24 asi_version 2-61
2.5.2.25 asi_dia 2-62
2.5.2.26 asi_dia_reset 2-64
2.5.2.27 asi_device_info 2-65
2.5.2.28 asi_cfg_in 2-66
2.5.2.29 asi_cfg_out 2-67
2.5.2.30 asi_in_all 2-68
2.5.2.31 asi_wd 2-69

2.6 Systemspezifika unter MS-DOS und Windows 2-70
2.6.1 Allgemeines 2-70
2.6.2 Ablage von Variablen 2-70
2.6.3 Laufwerks- und Pfadangaben 2-71
2.6.4 Anwenderprogramme an- und abmelden 2-71

2.7 Übersetzen und Binden für MS-DOS 2-72

2.8 Übersetzen und Binden für Windows 2-74

2.9 Besonderheiten für Windows 2-76

2.10 Hinweise und Fehlermeldungen 2-79
2.10.1 Das DPRAM 2-79
2.10.2 Status- und Fehlermeldungen des AS-I-Treibers 2-80
2.10.3 Status- und Fehlermeldungen des AS-I-Monitors 2-81
2.10.4 Returnwerte der AS-I-Aufrufe 2-83

A ASI PROTOCOL IMPLEMENTATION CONFORMANCE STATEMENT (PICS) A-1

A.1 PICS für CP 2413 A-1

B LITERATURVERZEICHNIS B-1

C ABKÜRZUNGEN / BEGRIFFE C-1

1 Einführung

Das vorliegende Gerätehandbuch gibt neben den allgemeinen Hinweisen eine Einführung in das SINEC S1-Systemkonzept und die SINEC S1-Komponenten. Der Schwerpunkt dieses Handbuchs liegt auf der Beschreibung der SINEC S1-Masterkomponenten.

Wir empfehlen Ihnen folgendes Vorgehen, wenn ...

...Sie sich einen Überblick über die Gesamtthematik SINEC S1 verschaffen wollen.	➥ Lesen Sie zunächst das Handbuch 'SINEC S1/AS-I Einführung und Grundlagen'. Dort finden Sie allgemeine Informationen zu SINEC S1.
...Sie wissen möchten, wie der CP 2413 zu programmieren ist.	➥ Lesen Sie im Handbuch 'SINEC S1/ AS-I Einführung und Grundlagen' im Kapitel 1 den Abschnitt 1.5 'Der Masterbetrieb'.Lesen Sie außerdem im vorliegenden Handbuch das Kapitel 2.
...Sie die SPS-Masterbaugruppe CP 2413 in Betrieb nehmen wollen.	➥ Im Kapitel 2 finden Sie unter den Abschnitten 2.3 'Übersicht CP 2413' und Abschnitt 2.4 'Softwareinstallation' schnell die nötigen Informationen.

1.1 Allgemeines

1.1.1 Kapitelübersicht

Kapitel 1 SINEC S1/AS-I Einführung

Das vorliegende Kapitel 1 enthält Hinweise für die effektive Nutzung dieses Handbuchs.

Kapitel 2 SINEC S1 Masterbaugruppe CP 2413

Das Kapitel beschreibt den Einsatz des CP 2413 im PC AT. Es wird der Funktionsumfang der Baugruppe beschrieben.

1.1.2 Allgemeine Lesehinweise und Schreibregeln

Allgemeine Symbole im Text:

✔ Dieses Zeichen fordert Sie zu einer Handlung auf.

☞ Dieses Zeichen macht Sie auf Besonderheiten und Gefahren aufmerksam.

1.1.3 Voraussetzungen

Voraussetzung zum Verständnis der kompletten Unterlage ist:

- Kenntnis des Handbuchs ‘SINEC S1/AS-I Einführung und Grundlagen’

1.1.4 Weitere Informationen

Bestellnummern zu den in diesem Handbuch erwähnten Produkten finden Sie im jeweilig aktuellen SINEC-Katalog.

Falls Sie noch weitere Informationen zu den SINEC S1- und AS-I-Komponenten von SIEMENS benötigen, wenden Sie sich bitte an Ihre zuständige SIEMENS Geschäftsstelle.

1.1.5 Hotline

Bitte wenden Sie sich bei technischen Rückfragen an folgende Hotline-Nummern:

09131-7-43147

09131-7-43157

2 Masterbaugruppe CP 2413 und AS-I-2413 MS-DOS/Windows

Das vorliegende Kapitel erläutert Ihnen die Leistungen der SINEC S1 Masterbaugruppe CP 2413 und macht Sie im Umgang mit der Baugruppe und der Schnittstelle AS-I-2413 MS-DOS/Windows vertraut.

Sie erfahren,

- wie die Baugruppe zu installieren ist,
- wie die Software auf Ihrem PC zu installieren ist,
- wie der Funktionsablauf im Anwenderprogramm zu implementieren ist,
- welche Funktionsaufrufe insgesamt zur Verfügung stehen,
- wie mit Fehlersituationen umzugehen ist.

Einordnung des Produktes

Mit der Baugruppe CP 2413 und der zugehörenden C-Schnittstelle AS-I-2413 MS-DOS/Windows erhält ein PC-AT Anschluß zu SINEC S1 in der Funktion als AS-I-Master. Wie folgende Darstellung zeigt, besteht im Verbund mit den weiteren verfügbaren CP-Baugruppen eine volle Integration in SINEC Netzen.

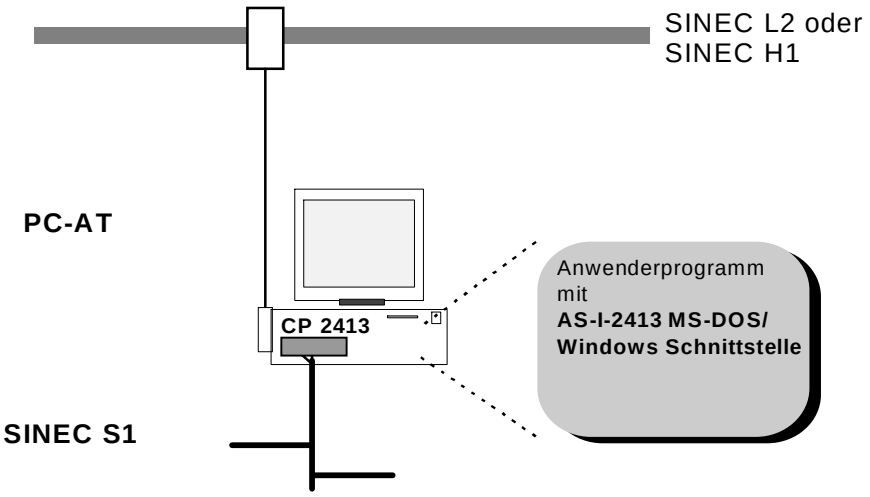


Bild 2-1: Einordnung des PC-AT mit CP 2413 Baugruppe

2.1 Übersicht AS-I-2413 MS-DOS/Windows

Die Schnittstellensoftware arbeitet nach folgendem Schema:

Anwenderprogramme (AS-I-Applikation) rufen die AS-I-Bibliotheksfunktionen auf. Diese benutzen Betriebssystemaufrufe, um mit dem AS-I-Treiber zu kommunizieren. Die Anschaltung wird über I/O-Adressen angesprochen, der Datenaustausch erfolgt über ein Dual-Port-RAM (DPRAM).

Folgendes Bild zeigt die genannten Komponenten von "AS-I-2413 MS-DOS/Windows".

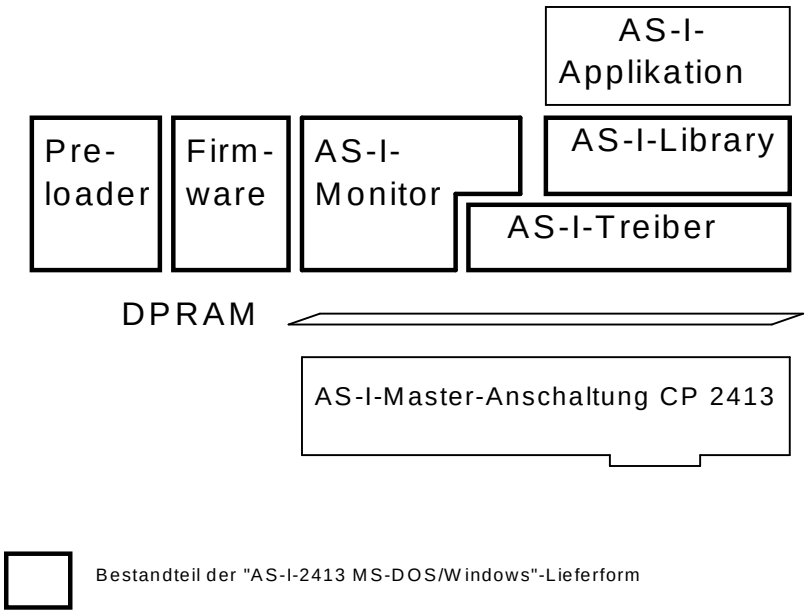


Bild 2-2 Komponenten von "AS-I-2413 MS-DOS/Windows"

- Der AS-I-Treiber:

  Der Treiber reicht die Daten vom Anwenderprogramm zum CP und umgekehrt. Er bietet keine direkte Schnittstelle zum Anwenderprogramm.

- Der AS-I-Monitor:

  Der AS-I-Monitor dient zum Konfigurieren des Treibers und übernimmt weitere, administrative Aufgaben.

- Die AS-I-Library:

  Die AS-I-Library enthält die AS-I-Schnittstelle und realisiert den Datenaustausch zwischen den Anwenderapplikationen und dem AS-I-Treiber. Anwenderapplikationen nutzen die AS-I-Library zur Kommunikation.

- Der Preloader:

  Der Preloader wird vom AS-I-Monitor auf die Baugruppe geladen und kopiert mit Hilfe des Monitors die zum Betrieb notwendige Firmware auf die Anschaltung.

- Die Firmware:

  Die Firmware auf dem CP 2413 wickelt den Datenaustausch mit den Slaves ab.

2.2 Ablaufumgebung und Anforderungen

Nachdem Sie in den vorigen Kapiteln Einsatzbereiche, Schnittstellen und Komponenten von "AS-I-2413 MS-DOS/Windows" kennengelernt haben, werden Sie in diesem Kapitel erfahren, welche Anforderungen dieses Produkt an Ihren Rechner stellt.

Neben den in diesem Dokument beschriebenen SINEC Komponenten (Kommunikationsprozessor CP 2413 und SINEC Software Module) muß für die Kommunikation ein SINEC S1-Netz zur Verfügung stehen.

Weiterhin gelten für den Rechner, auf dem der CP 2413 eingesetzt werden soll, die folgenden Hard- und Softwareanforderungen (siehe hierzu auch beiliegende Produktinformation und die Datei 'readme.txt' auf der ersten Installationsdiskette).

2.2.1 Hardwareanforderungen

Als Hardwarevoraussetzung benötigen Sie einen AT-kompatiblen Rechner mit einem 2kB großen Adreßbereich, der im Hauptspeicher frei verfügbar sein muß (Anfangsadresse: C8000h, C8800h, C9000h ... EF800h) sowie freie I/O-Ports für die Konfigurationsregister des CP. Diese Konfigurationsregister können über einen DIL-Schalter eingestellt werden (Basiswerte: 200h, 208h, 280h, 288h, 290h, 320h, 3e0h, 3e8h).

Falls mehrere Adreßbereiche und I/O-Ports frei sind, so ist der gleichzeitige Betrieb von bis zu vier CP 2413-Baugruppen möglich. Sind darüberhinaus die Voraussetzungen zum Betrieb eines CP 1413 bzw. eines CP 5412 erfüllt, so ist auch der Parallelbetrieb dieser Baugruppen mit AS-I-Master-Anschaltungen möglich.

☞ **Lesen Sie hierzu die Hinweise zu den CP 1413- bzw. CP 5412-Produkten wie "TF-NET1413/MS-DOS, Windows", "MAP-NET1413/MS-DOS, Windows" und "TF-NET5412/MS-DOS, Windows".**

Eine Liste der Rechner, auf denen der CP 2413 betrieben werden kann, entnehmen Sie bitte der Datei 'readme.txt' auf der ersten Installationsdiskette.

2.2.2 Softwarevoraussetzungen

In der Produktinformation und der Datei 'readme.txt' sind die Betriebssystemversionen von MS-DOS und Windows aufgeführt, die Voraussetzung für den Betrieb des CP 2413 sind.

2.3 Übersicht CP 2413

Sie kennen nun die Anforderungen, die "AS-I-2413 MS-DOS/Windows" an Ihren Rechner stellt. In diesem Abschnitt erfahren Sie, wie Sie die Baugruppe CP 2413 einstellen und in ihren Rechner einbauen müssen.

2.3.1 Hardwareinstallation

Der CP 2413 benötigt einen AT-Slot. Auf der Baugruppe ist lediglich das Basis-Konfigurationsregister über einen DIL-Schalter einzustellen. Die Lage dieses Schalters ist Bild 2-3 zu entnehmen. Der Wert des Basis-Konfigurationsregisters ist per Default 0x0320.

Neben dem Defaultwert sind noch die im folgenden Bild wiedergegebenen Schalterstellungen möglich.

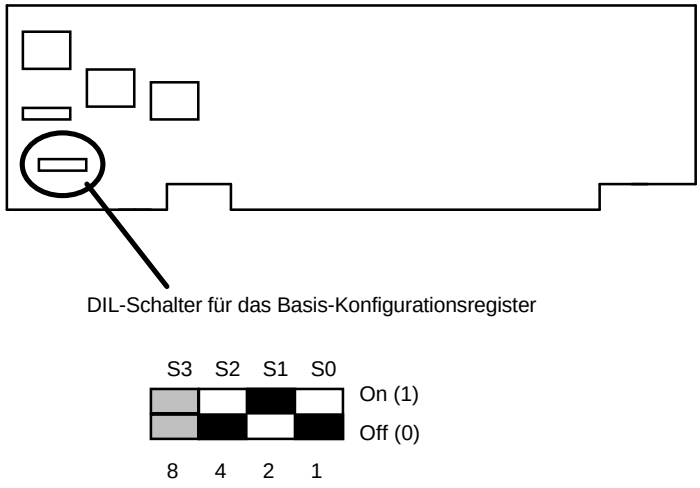


Bild 2-3 Grundriß des CP 2413

Über einen 8 Byte großen I/O-Adressraum, beginnend mit einem der in Bild 2-4 dargestellten Basis-Konfigurationsregister, werden sämtliche, für den Betrieb der Baugruppe notwendigen Werte per Software auf den CP geladen. In der Grundeinstellung werden die I/O-Ports 320h bis 327h belegt.

S3	S2	S1	S0	Konf.-Reg.
-	0	0	0	0200H
-	0	0	1	0208H
-	0	1	0	0320H
-	0	1	1	03E0H
-	1	0	0	03E8H
-	1	0	1	0280H
-	1	1	0	0288H
-	1	1	1	0290H

Bild 2-4 Basis-Konfigurationsregister

Einbauhinweis

☞ **Der Einbau der Baugruppe in den Rechner darf nur in spannungslosem Zustand erfolgen. Beim Hantieren der Baugruppe sind die üblichen Schutzmaßnahmen gegen Gefährdung durch elektrostatische Aufladung zu treffen.**

Informationen über das Öffnen des betreffenden Geräts, Wahl des Steckplatzes für die Anschaltung und Montageanleitung für Baugruppen im vorliegenden Format sind der jeweiligen Betriebsanleitung Ihres PGs oder PCs zu entnehmen.

2.3.2 Technische Daten

Die wichtigsten technischen Daten entnehmen Sie bitte der folgenden Darstellung und der Tabelle.

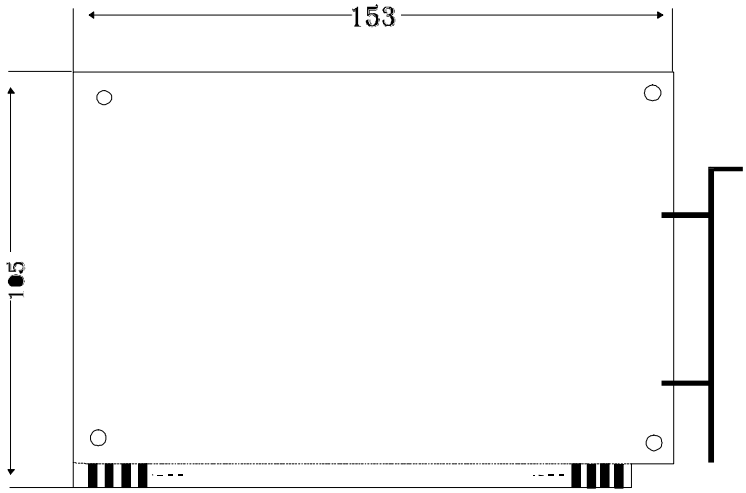


Bild 2-5 Maßzeichnung CP 2413

Buszykluszeit	5 ms bei 31 Slaves
Datentransfer PC - CP 2413	über 2 Kbyte Dual-Port-RAM
Schnittstellen	
• Anschluß AS-I	über Stecker mit Klemmenanschluß an AS-I-Profilleitung
Versorgungsspannung Zulässige Spannung an der AS-I-Leitung	DC 5V über Rückwandbus
Zulässige Umgebungsbedingungen	
• Betriebstemperatur	0..55°C
• Transport- und Lagertemperatur	-40°C bis +70°C
• Relative Feuchte	95% bei +25°C
Stromaufnahme Verlustleistung der Baugruppe typ.	typ. 720 mA
Konstruktiver Aufbau	
• Baugruppenformat	Flachbaugruppe, kurzes AT-Format
• Maße (B x H x T) in mm	107 x 152
• Gewicht ca.	110g
• Platzbedarf	1 SEP (15,24 mm)

Tabelle 2-1 Technische Daten

2.3.3 Anschluß der AS-I-Leitung

Der Anschluß der AS-I-Leitung erfolgt über den mitgelieferten Stecker, der über zwei intern im CP 2413 gebrückte Klemmenpaare verfügt.

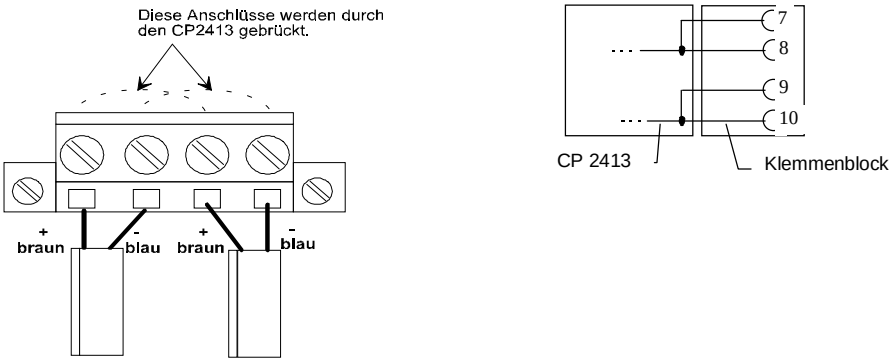


Bild 2-6 Stecker für den Anschluß der AS-I-Leitung

Das zweite Klemmenpaar ist für den Anschluß des AS-I-Netzteils oder einen Abzweig der AS-I-Leitung vorgesehen. Das AS-I-Netzteil kann jedoch grundsätzlich auch an jeder anderen Stelle der AS-I-Leitung angeschlossen werden.

2.4 Softwareinstallation

Im vorigen Kapitel haben Sie erfahren, wie Sie den CP 2413 für "AS-I-2413 MS-DOS/Windows" einstellen und in Ihrem Rechner einbauen müssen. Jetzt erfahren Sie, was Sie tun müssen, um die zum Betrieb erforderliche Software auf Ihrem Rechner zu installieren.

☞ **Lesen Sie bitte vor der Inbetriebnahme die Produktinformation und die Datei 'readme.txt' auf der ersten Lieferdiskette.**

2.4.1 Was Sie zur Installation brauchen

Rechner

Dieses Produkt kann auf den Programmiergeräten PG 770, PG 740 und kompatiblen PCs eingesetzt werden.

Betriebssystem

Das Produkt kann sowohl unter MS-DOS ab Version 5.0, als auch unter Windows 3.x im enhanced-Modus betrieben werden.

CP 2413

Zum Betrieb benötigen Sie einen SINEC-Kommunikationsprozessor CP 2413. Diese Anleitung geht davon aus, daß Sie den CP nach Anleitung eingestellt und bereits in Ihren Rechner eingebaut haben. Beim gleichzeitigen Einbau von mehreren Baugruppen ist darauf zu achten, daß auf jedem CP eine andere I/O-Adresse eingestellt ist. Beachten Sie beim Einbau der CP 2413 die Hinweise in 2.3.1.

I/O-Port

Jeder CP 2413 belegt acht fortlaufende Portadressen ab einer einstellbaren Basisadresse. Als Basisadressen stehen zur Auswahl:

0x200, 0x208, 0x280, 0x288,
0x290, 0x320, 0x3E0, 0x3E8.

Stellen Sie bitte sicher, daß die von Ihnen gewählten Portadressen in Ihrem Rechner nicht anderweitig belegt sind.

Adreßbereich (Dual-Port-RAM)

Jeder CP 2413 belegt beim Betrieb einen Speicherbereich von 2Kbyte, dessen Anfangsadresse im Bereich C8000-EF800 liegen darf.

Stellen Sie bitte sicher, daß der von Ihnen gewählte Speicherbereich von keiner anderen Komponente des Rechners benutzt wird.

Installationsdisketten

Dieses Produkt wird auf Disketten im Format 8,89 cm (3,5“) geliefert.

- Die Software-Diskette(n) sind schreibgeschützt.
- Die Schlüsseldiskette enthält nur die Autorisierung und ist nicht schreibgeschützt.

2.4.2 Wie Sie vorgehen

Vorbereitung der SW-Installation

Zur Vorbereitung der Softwareinstallation gehen Sie bitte wie folgt vor:

- Lesen Sie die Datei README.TXT auf der ersten Lieferdiskette dieses Produkts.
- Lesen Sie diese Installationsanleitung sorgfältig durch.
- Lesen Sie die Hinweise in Kap. 2.3.1. zum CP 2413 und installieren Sie die Baugruppe gemäß der Richtlinie.

Installation der Software

Die Softwareinstallation kann unter MS-DOS und Windows erfolgen. Bei einer Installation unter MS-DOS gehen Sie wie folgt vor:

- Starten Sie Ihren Rechner so, daß Sie unter MS-DOS eine Eingabeaufforderung erhalten.
- Legen Sie in das Diskettenlaufwerk (in dieser Anleitung weiterhin mit a: bezeichnet) die erste Diskette der Lieferform ein.
- Wechseln Sie auf das Laufwerk mit der ersten Produktdiskette, z.B. durch Eingabe von

  a:
- Wechseln Sie auf die Root des Laufwerks durch Eingabe von ‘cd\’
- Starten Sie das Installationsprogramm durch Eingabe von:

  install
- Folgen Sie den Anweisungen des Installationsprogramms. Hinweis: Im Installationsprogramm besteht die Möglichkeit, ASI-2413 MS-DOS/Windows ohne Autorisierung zu installieren. In diesem Fall ist die Software nicht ablauffähig. Sie haben jedoch die Möglichkeit, nachträglich die Autorisierung zu installieren.

Bei einer Installation unter Windows ergibt sich der folgende Ablauf:

- Wählen Sie unter Windows im Programm-Manager die Schaltfläche „Datei/Ausführen“ an
- Legen Sie in das Diskettenlaufwerk (in dieser Anleitung weiterhin mit a: bezeichnet) die erste Diskette der Lieferform ein.
- Starten Sie das Installationsprogramm durch Eingabe von:

  a: setup
- Folgen Sie den Anweisungen des Installationsprogramms. Hinweis: Im Installationsprogramm besteht die Möglichkeit, ASI-2413 MS-DOS/Windows ohne Autorisierung zu installieren. In diesem Fall ist die Software nicht ablauffähig. Sie haben jedoch die Möglichkeit, nachträglich die Autorisierung zu installieren.

Konfiguration der Software

Nach der Übertragung der Software auf Ihre Festplatte startet SINEC-Setup automatisch und Sie können den CP 2413 konfigurieren.

- Wählen Sie in der Hardware-Auswahlliste (untere Auswahlliste) einen Eintrag für den CP-Typ „CP 2413“ aus.
- Rufen Sie über den Menüpunkt „Bearbeiten / Parameter ändern“ das Konfigurationsprogramm auf.
- Nehmen Sie die notwendigen Einstellungen vor. Beachten Sie dabei die Hinweise der Abschnitte Hardwareanforderungen und Hardwareinstallation des Kapitels 2.
- Beenden Sie das Konfigurationsprogramm und speichern Sie Ihre Konfiguration ab.
- Beenden Sie SINEC-Setup.

2.4.3 Was Sie beachten müssen

Autorisierung

Dieses Produkt wird mit einer Schlüsseldiskette (Autorisierungsdiskette) ausgeliefert. Auf dieser befindet sich ein Autorisierungsschlüssel, der für den Betrieb des Produkts notwendig ist. Bei der Installation wird dieser Schlüssel auf Ihrer Festplatte abgelegt. Der Betrieb des Produkts ohne den Autorisierungsschlüssel ist nicht möglich.

Über den Menüpunkt „Autorisierung/Übertragen“ können Sie mit dem SINEC-Setup die Autorisierungsinformation auch nachträglich installieren.

Die Autorisierungsinformation wird in dem versteckten Verzeichnis „\ax nf zz“ auf dem Laufwerk abgelegt, auf dem Sie das Produkt installiert haben.

Löschen Sie bitte nicht das Verzeichnis „\ax nf zz“ oder darin abgelegte Dateien!

Falls Sie Ihre Festplatte neu formatieren wollen, müssen Sie die Autorisierungsinformation vorher über den Menüpunkt „Autorisierung/ Rückübertragen“ des SINEC-Setup auf die Schlüsseldiskette zurück-übertragen. Andernfalls geht die Autorisierungsinformation verloren, und Sie können das Produkt nicht weiter nutzen. Stellen Sie bei der Rückübertragung sicher, daß Sie die Diskette eingelegt haben, von der Sie die Autorisierungsinformation installiert haben.

Es ist nicht möglich, die Autorisierungsinformation durch einfaches Kopieren von der Festplatte auf eine Diskette oder von der Diskette auf die Festplatte zu übertragen.

2.4.4 Hantierungsmöglichkeiten nach der Installation

2.4.4.1 SINEC-Setup

Das SINEC-Setup-Programm

Das SINEC-Setup-Programm faßt unter einer Oberfäche alle Hantierungsmöglichkeiten der installierten SINEC-Produkte zusammen.

Die Oberfläche von SINEC-Setup

Die Oberfläche von SINEC-Setup präsentiert sich Ihnen mit vier Bestandteilen:

- der Menüleiste,
- der Liste der installierten Softwarepakete,
- der Liste der installierten Hardwarekomponenten (Kommunikationsprozessoren) und
- der Zuordnungsmatrix von installierten Softwarepaketen und Hardwarekomponenten.

Die Menüleiste

Die Menüleiste von SINEC-Setup enhält die Menüpunkte:

- Datei
  mit der Funktion Beenden, mit der Sie SINEC-Setup verlassen können,
- Bearbeiten
  mit allen für die ausgewählte Hard- oder Software verfügbaren Funktionen,
- Extra
  mit den möglichen Spracheinstellungen und
- Hilfe
  mit dem Zugang zur Online-Hilfe.

Die Liste der installierten Softwarepakete

In der Liste der installierten Softwarepakete sind alle neuen SINEC-Softwarepakete aufgeführt, welche auf demselben Laufwerk installiert wurden.

Die Liste der installierten Hardwarekomponenten

In der Liste der installierten Hardwarekomponenten sind alle SINEC-CPs aufgeführt, deren Konfigurationsdaten auf demselben Laufwerk installiert wurden.

Es werden dabei auch CPs erfaßt, welche ohne SINEC-Setup installiert wurden (Ausnahme CP2413). Bei ihnen kann über das „Bearbeiten“-Menü das Programm netinst.exe aufgerufen werden.

Die Zuordnungsmatrix

Durch die Zuordnungsmatrix haben Sie einen schnellen Überblick darüber, welches Softwarepaket mit welcher Hardware eingesetzt werden kann, und welches Softwarepaket von Ihnen gerade auf welcher Hardware eingesetzt wird.

Die mögliche Kombination eines Softwarepakets mit einer Hardwarekomponente ist durch „-“ gekennzeichnet.

Ein auf einer Hardware aktiviertes Softwarepaket ist durch „x“ gekennzeichnet.

2.4.4.2 Autorisierung übertragen und prüfen

Autorisierung

Zu jedem Softwarepaket gehört eine Autorisierung, welche es Ihnen ermöglicht, die Software zu betreiben.

Unter dem „Bearbeiten“-Menü erscheint bei angewähltem Softwarepaket der Menüpunkt „Autorisierung“ mit den Unterfunktionen „Übertragen“, „Rückübertragen“ und „Prüfen“.

Übertragen

Bei dieser Funktion werden Sie gefragt, von welchem Laufwerk zu welchem Laufwerk Sie die Autorisierungsinformation übertragen wollen. Die Übertragung erfolgt stets von der Schlüsseldiskette auf die Festplatte.

Bitte unterbrechen Sie den Übertragungsvorgang nicht, die Autorisierungsinformation könnte sonst verloren gehen.

Rückübertragen

Bei dieser Funktion werden Sie gefragt, von welchem Laufwerk Sie die Autorisierungsinformation rückübertragen wollen. Die Übertragung erfolgt stets von der Festplatte auf die Schlüsseldiskette.

Bitte unterbrechen Sie den Übertragungsvorgang nicht, die Autorisierungsinformation könnte sonst verloren gehen.

Prüfen

Bei dieser Funktion werden Sie gefragt, welches Laufwerk Sie auf das Vorhandensein der Autorisierungsinformation für das angewählte Produkt überprüfen wollen.

Sie können mit dieser Funktion sowohl Ihre Festplatte, als auch eine Diskette auf das Vorhandensein der Autorisierungsinformation untersuchen.

2.4.4.3 Konfiguration des CP 2413

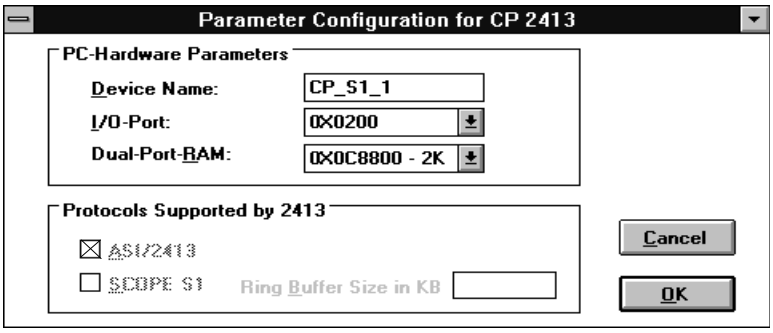
Ablauf der Konfigurierung

Gehen Sie wie folgt vor, um einen CP 2413 zu konfigurieren:

Wählen Sie den CP 2413 im Setup durch Anklicken der entsprechenden Zeile in der Auswahlliste aus.

Durch Anwahl des Menüpunktes „Bearbeiten/Parameter ändern“ (Ändern der Parameter des CPs) oder „Bearbeiten/Neuen CP konfigurieren“ (Erstellen eines Satzes von Konfigurationsparametern für einen weiteren CP) wird die Eingabemaske für die Konfigurierung des CP aufgerufen.

Über den Menüpunkt „Bearbeiten/Konfigurierten CP löschen“ kann ein Satz von Konfigurationsparametern wieder gelöscht werden.



In der Parametriermaske können folgende Eingaben erfolgen:

Devicename

Kennzeichnet, für welche Anwendung der CP 2413 benutzt wird. Der Name ist frei wählbar. Mit diesem Namen kann eine Anwenderapplikation den CP ansprechen (siehe hierzu z.B. Kapitel 2.5.2.1).

Bis zu vier CP 2413 können für die Anwendung ASI-Master parallel betrieben werden. Für diese CP 2413 müssen unterschiedliche Device-Namen, I/O-Port- und Dual-Port-RAM-Adressen ausgewählt werden.

I/O-Port

Hier müssen Sie die beim CP 2413 eingestellte I/O-Adresse auswählen. Der ausgewählte Wert muß mit der Schaltereinstellung auf dem CP 2413 übereinstimmen (siehe hierzu auch die Produktinformation des CP 2413). Stellen Sie sicher, daß diese I/O-Adresse nicht bereits von einer anderen PC-Karte benutzt wird.

Dual-Port-RAM

Durch Eingabe der Dual-Port-RAM-Adresse wählen Sie aus, über welchen Speicherbereich der CP 2413 mit Ihrem PC kommuniziert. Der CP 2413 belegt 2Kbyte im Adreßraum Ihres PCs. Auch hier müssen Sie sicherstellen, daß dieser Adreßbereich nicht bereits vom PC selbst oder durch eine andere PC-Karte belegt ist.

Auswahlbox

Durch Anklicken von ASI-2413 in der Auswahlbox legen Sie fest, daß der CP 2413 für die Anwendung AS-I-Master verwendet wird. Durch nochmaliges Anklicken wird der CP 2413 für ASI-2413 deaktiviert und kann z.B. für die Anwendung SCOPE S1 benutzt werden.

Beenden des Konfigurationsprogrammes

Erfolgt nach dem Beenden des Konfigurationsprogrammes die Meldung „Reboot your PC after leaving Setup to activate all indicated CPs/devices“, dann muß Ihr PC neu gebootet werden, damit die durchgeführten Änderungen übernommen werden.

Beachten Sie, daß ein CP 2413 nur dann für ASI-2413 MS-DOS/Windows benutzbar ist, wenn er zuvor wie hier beschrieben konfiguriert wurde.

Die Datei autoexec.bat

Das Installationsprogramm erweitert die Datei autoexec.bat ihres Rechners um den Eintrag „call <lw:>\sinec\sinecini.bat“. Dieser Eintrag wird zur Initialisierung der CP 2413 Baugruppen in Ihrem Rechner benötigt. Ihre ursprüngliche Datei autoexec.bat wird als autoexec.bak gesichert.

Ändern der Konfiguration

Sie können nachträgliche Änderungen Ihrer Konfiguration durch erneutes Starten von SINEC Setup vornehmen. Setup steht sowohl unter MS-DOS als auch unter Windows zur Verfügung, falls die Installation unter Windows erfolgt ist.

Setup unter DOS bzw. unter Windows wird durch Aufruf von setup.exe gestartet. Dieses Programm befinden sich im Verzeichnis <lw:>\SINEC <lw:> ist das Laufwerk, auf dem Sie ASI-2413 MS-DOS/Windows installiert haben.

Das Installationsprogramm unter Windows legt in der Defaulteinstellung eine SINEC-Programmgruppe an, in der eine Ikone für SINEC Setup enthalten ist. Sie können SINEC Setup in diesem Fall auch durch Doppelklicken auf diese Ikone starten.

2.4.5 Verzeichnisstruktur

Die für ASI-2413 MS-DOS/Windows benötigten Dateien werden bei der Installation im Verzeichnis

<lw:> SINEC

abgelegt. („lw“ ist das bei der Installation angegebene Zielverzeichnis)

2.4.6 Initialisieren des CP2413, die Start-Batch-Datei

Bei der Installation von ASI-2413 MS-DOS/Windows wird in die Datei autoexec.bat vom Installationsprogramm der Eintrag call \sinec\sinecini.bat eingefügt. Durch den Aufruf der Batch-Datei „sinecini.bat“ im Verzeichnis SINEC wird die Betriebsbereitschaft der AS-I-Masteranschaltungen CP 2413 hergestellt.

Die Datei sinecini.bat wird durch das Konfigurierungsprogramm im Setup aktualisiert. Sie ruft wiederum Batchdateien auf, die die einzelnen SINEC CPs initialisieren (Die Datei ini2413.bat initialisiert z.B alle konfigurierten CP 2413-Baugruppen in Ihrem Rechner).

2.4.7 AS-I-Treiber

Der AS-I-Treiber wird bei der Installation in das Directory '\sinec\cp2413.dw\bin' auf Ihren Zielrechner kopiert. Er ist als speicherresidentes Programm realisiert, das beim Anlauf eine Interruptroutine in die Interrupt-Sprungleiste einhängt, über die der gesamte Handshake mit bis zu vier AS-I-Master-Baugruppen abgewickelt wird.

Der AS-I-Treiber wird automatisch durch die Batchdateien sinecini.bat bzw. ini2413.bat aufgerufen, so daß Sie keine zusätzlichen Aufrufe für die Treiberinitialisierung absetzen müssen. Der Vollständigkeit wegen wird dennoch die Aufrufsyntax des Treibers erläutert.

Für den Aufruf des AS-I-Treibers gilt folgende Syntax:

asidrv.exe x	installiert den Treiber, wobei 'x' als Hex-Wert interpretiert wird und die Interruptnummer darstellt. Zulässig sind alle geradzahligen Hex-Werte von 80h bis einschließlich c0h. Beispielsweise wird durch den Aufruf 'asimon 80' der AS-I-Treiber installiert unter Nutzung der Interruptnummer 80h.
asidrv.exe	sucht einen freien Interrupt im obigen Bereich und installiert den Treiber.

Die Status- und Fehlermeldungen des AS-I-Treibers sind in Kapitel 2.10.2 zusammengefaßt, einschließlich einiger Lösungsvorschläge für eventuell auftretende Probleme.

2.4.8 AS-I-Monitor

Der AS-I-Monitor wird bei der Installation in das Directory '\sinec\cp2413.dw\bin' auf Ihren Zielrechner kopiert. Er dient dazu, den AS-I-Treiber zu konfigurieren, administrative Aufträge, wie z.B. das Laden der Firmware auf die Baugruppe, abzuhandeln und die AS-I-Master-Baugruppe in einen betriebsbereiten Zustand zu bringen. Im Gegensatz zum AS-I-Treiber, der bis zu vier Baugruppen gleichzeitig unterstützt, muß der AS-I-Monitor für jede AS-I-Master-Baugruppe einmal mit den für die Anschaltung aktuellen Parametern aufgerufen werden.

Der AS-I-Monitor wird automatisch durch die Batchdateien sinecini.bat bzw. ini2413.bat aufgerufen, so daß Sie keine zusätzlichen Aufrufe für die Treiberinitialisierung absetzen müssen. Der Vollständigkeit wegen wird dennoch die Aufrufsyntax des Treibers erläutert.

Für den Aufruf des AS-I-Monitors gilt folgende Syntax:

asimon <option>

Für <option> sind die aufgeführten Werte definiert:

-?	Es wird eine Übersicht der möglichen Optionen und ihre Bedeutung auf dem Bildschirm ausgegeben.
-h	Die Kommunikation über sämtliche AS-I-Master-Anschaltungen wird beendet. Die Baugruppen werden zurückgesetzt, der AS-I-Treiber wird aus dem Speicher entfernt und der von ihm belegte Interrupt freigegeben.
-Dw -cx -dy -ez	Hiermit wird der AS-I-CP mit dem Namen 'w' gekennzeichnet. Dieser Name dient zur Identifizierung der AS-I-Master-Anschaltung und darf maximal 8 Zeichen lang sein. Der Treiber wird konfiguriert, die in der Datei '\sinec\cp2413.hw\fw2413' gesicherte Firmware im mit 'x' gekennzeichneten Laufwerk auf die Anschaltung geladen und der CP gestartet. Die Anschaltung belegt das DPRAM mit der Basissegmentadresse 'y' und das Basis-Konfigurationsregister mit dem Wert 'z'. Die Reihenfolge dieser Optionen ist zwingend.

Beispiel:

-DCP_ASI1: -cc: -dd000 -e0320

Bei Aufruf des Monitors mit diesen Optionen in der angegebenen Reihenfolge wird eine AS-I-Master-Anschaltung selektiert, die mit dem Namen 'CP_ASI1:' angesprochen wird. Die Firmware-Datei '\sinec\cp2413.hw\fw2413' auf Laufwerk 'c:' wird auf die Anschaltung geladen. Die Baugruppe benutzt das DPRAM ab Adresse d000:0 und den I/O-Adressraum von 0320h bis 0327h.

Die Status- und Fehlermeldungen des AS-I-Monitors sind in Kapitel 2.10.3 zusammengefaßt, einschließlich einiger Lösungsvorschläge für eventuell auftretende Probleme.

2.4.9 Die AS-I-Library

Die AS-I-Library wird bei der Installation in das Directory '\sinec\cp2413.dw\lib' auf Ihren Zielrechner kopiert. Sie liegt für verschiedene Speichermodelle, Compiler und Betriebssysteme vor und ist Voraussetzung für Applikationen, die mit der AS-I-Master-Anschaltung bzw. über die Baugruppe mit Slaves kommunizieren wollen.

Nähere Informationen zur Schnittstelle finden Sie in Kapitel 2.5, Vorschriften zum Binden Ihrer Applikationen in Kapitel 2.7 bis 2.9

2.4.10 Beispielprogramme

Beispielprogramme werden bei der Installation in das Directory '\sinec\cp2413.dw\samples' auf Ihrem Zielrechner kopiert. Sie liegen als Quellcode vor und können mit Hilfe der mitgelieferten Generieranweisungen übersetzt und gebunden werden. Unter Umständen ist dabei eine Anpassung der Generieranweisungen an Ihre Systemumgebung vorzunehmen. Hinweise auf die Funktionsweise der Beispiele finden Sie in den Help-Texten.

2.5 Die AS-I-Anwenderschnittstelle

2.5.1 Wirkungsweise der Aufträge

Zum besseren Verständnis der einzelnen Bibliotheksfunktionen wird eine kurze Einführung in die Funktionsweise der AS-I-Master-/Slaveanordnung mit dem CP 2413-Master gegeben. Im nachfolgenden Bild ist als Basis für die weiteren Erläuterungen eine Master-Slave-Anordnung aufgezeigt.

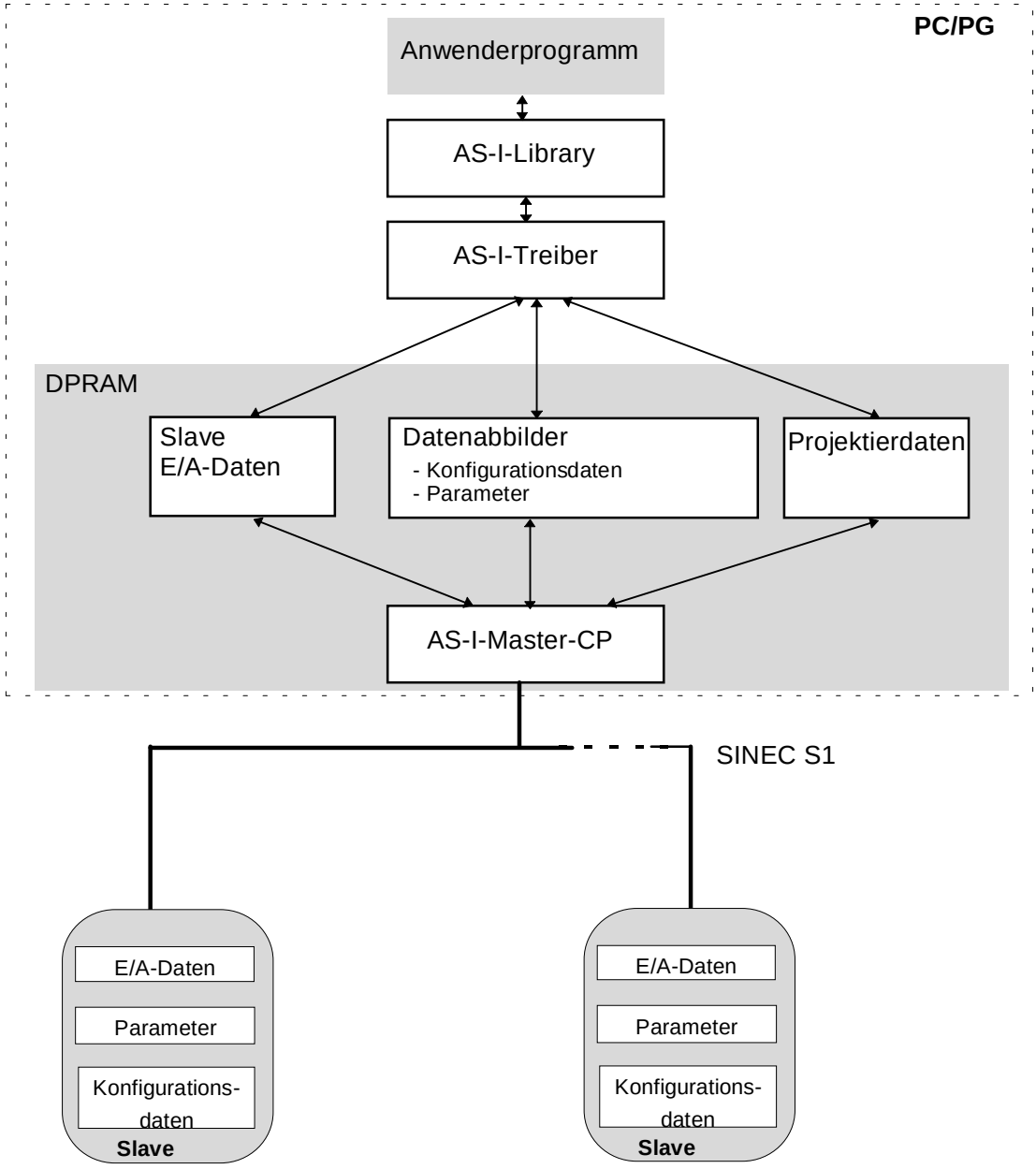


Bild 2-7 Datenorganisation Master-Slave

Anlauf

Im Anlauf versucht der AS-I-Master, alle an der AS-I-Leitung befindlichen und funktionsfähigen ASI-Slaves zu finden (Erkennungsphase). Dazu liest er die E/A-Konfiguration und den Identifikationscode aller Slaves (d.h. durch Senden der entsprechenden Telegramme an die erlaubten Adressen von 0 bis 31). Ein AS-I-Slave gilt nur dann als erkannt und wird nur dann in die Liste der erkannten Slaves aufgenommen, wenn beide Aufrufe erfolgreich durchgeführt werden konnten. Diese Liste (LDS) kann mit der Library-Funktion **'asi_get_lds()**' ausgelesen werden.

Aktivierungsphase

Danach versucht der AS-I-Master, die bereits erkannten Slaves in den zyklischen Datenaustausch und damit in die Liste der aktiven Slaves (LAS) aufzunehmen, indem die Slaves durch Schreiben der Parameterausgänge aktiviert werden (Aktivierungsphase).

Normalbetrieb

Daran schließt sich der Normalbetrieb des AS-I-Masters an, der sich in die drei folgenden Phasen gliedern läßt, die sich zyklisch wiederholen:

- Datenaustauschphase:

  In der Datenaustauschphase werden an sämtliche Slaves aus der Liste der aktiven Slaves die jeweiligen Ausgangsdaten übertragen. Die angesprochenen Slaves übermitteln bei fehlerfreier Übertragung die Werte der Slaveeingänge. Die Ausgangsdaten für die Slaves werden aus einem Abbild im DPRAM entnommen, das durch die Funktion **'asi_write_odi()**' jederzeit von der übergeordneten Steuerungsapplikation an die jeweiligen Erfordernisse angepaßt werden kann. Die Werte der Slaveeingänge werden in einem Abbild im DPRAM abgelegt, auf das mit Hilfe der Funktion **'asi_read_idi()**' lesend zugegriffen werden kann.

- Managementphase:

  In dieser Phase werden Aufträge der übergeordneten Steuerungsapplikation an die am Bus angeschlossenen und aktivierten Slaves abgearbeitet.

- Aufnahmephase:

  In der Aufnahmephase werden neu hinzukommende Slaves in die Liste der erkannten und der aktiven Slaves aufgenommen. Die durchgeführten Maßnahmen entsprechen denen im Anlauf. Darüberhinaus erfolgt eine Wiederaufnahme temporär ausgefallener Slaves.

Der AS-I-Master unterscheidet zwei Betriebsmodi, die mit der Funktion **'asi_set_operation_mode()'** eingestellt werden können:

- Projektierungsmodus

  Im Betriebsmodus 'Projektierungsmodus' werden alle AS-I-Slaves, die vom AS-I-Master am Bus erkannt werden, in den zyklischen Datenaustausch aufgenommen. Gedacht ist dieser Modus für eine erste Inbetriebnahme des AS-I-Masters oder für eine Umprojektierung. Hier kann die Funktionsweise der einzelnen Komponenten getestet und gegebenenfalls schnell verändert werden. Die Versorgung der Slaves mit Parameterwerten erfolgt mit Hilfe der Funktion **'asi_write_parameter()**'. Nach Abschluß der Umprojektierung und der Tests können die so ermittelten und eingestellten Werte 'eingefroren', d.h. projektiert werden. Dies geschieht durch eine nichtflüchtige Speicherung auf der AS-I-Master-Anschaltung.

- Geschützter Betrieb

  Im Betriebsmodus 'geschützter Betrieb' sucht der AS-I-Master nach allen am Bus befindlichen AS-I-Slaves. Nur wenn deren Konfigurationsdaten mit denen der Projektierung übereinstimmen, werden sie in den zyklischen Datenaustausch aufgenommen.

Zu den Projektierdaten gehören:

- die Liste der projektierten Slaves (LPS)
  Sie gibt an, welche AS-I-Slaves am AS-I-Bus angeschlossen sein sollten, und wird mit der Funktion 'asi_set_lps()' auf der AS-I-Master-Anschaltung gespeichert.

- die E/A-Konfiguration und die Identifikationscodes der AS-I-Slaves
  Diese Werte werden per Funktionsaufruf **'asi_set_permanent_configuration()**' auf der AS-I-Master-Anschaltung gesichert.

- Parameter, die beim Neustart des AS-I-Masters zu den AS-I-Slaves übertragen werden
  Diese Parameter werden mit der Funktion **'asi_set_permanent_parameter()**' eingestellt.

- verschiedene Flags
  Neben dem Flag 'Betriebsmodus' kann noch ein weiteres Flag gesetzt oder zurückgenommen werden. Mit dem Aufruf '**asi_set_offline_mode()'** wird zwischen dem Online- und dem Offline-Betrieb umgeschaltet. Im Online-Betrieb gliedert sich der Normalbetrieb der AS-I-Master-Anschaltung in die drei zuvor beschriebenen Phasen Datenaustauschphase, Managementphase und Aufnahmephase. Im Offline-Betrieb werden lediglich Aufträge von der übergeordneten Steuerungsapplikation entgegengenommen. Ein zyklischer Datenaustausch und die Aufnahme von neuen Slaves entfällt.

Neben diesen zur Projektierung notwendigen Funktionen stellt die AS-I-Library noch zwei weitere Funktionen zur Verfügung:

- Aktuelle Parameter projektieren
  Mit Hilfe der Funktion **'asi_store_actual_parameters()**' können ebenfalls Parameter projektiert werden. Allerdings werden die zu projektierenden Parameter nicht vom Aufrufer an den AS-I-Master übergeben, sondern es werden die aktuellen Parameter verwendet.

- Aktuelle Konfiguration projektieren
  Mit Hilfe der Funktion **'asi_store_actual_configuration()**' werden die projektierten Konfigurationsdaten und Identifikationscodes mit den aktuellen Werten der Slaves an der AS-I-Leitung überschrieben.

Parameter- und Konfigurationsdaten auslesen

Die projektierten Parameter und Konfigurationsdaten sowie die aktuellen Parameter können aus einem Abbild im DPRAM ausgelesen werden. Dies wird von der AS-I-Library durch folgende Funktionen unterstützt:

- Die Funktionen **'asi_get_lps()', 'asi_get_permanent_parameter()**', **'asi_get_permanent_configuration()**' und **'asi_read_parameter()**' übergeben die Liste der projektierten Slaves, die projektierten Parameter, Konfigurationsdaten und Identifikationscodes bzw. die aktuellen Parameter an den Aufrufer.

- Die Funktion **'asi_get_flags()**' liest sämtliche Flags aus.

Zur Unterscheidung der AS-I-Slaves am Bus besitzt jeder Slave eine Adresse im Bereich von 0 bis 31, unter der ihn der AS-I-Master ansprechen kann.

Der AS-I-Master kann mit Hilfe der Funktion **'asi_change_slave_address()'** jedem AS-I-Slave eine neue Adresse zuweisen. Der AS-I-Slave mit der Adresse 0 nimmt hierbei eine Sonderstellung ein, denn nur ihm kann eine von 0 verschiedene Adresse durch den AS-I-Master zugewiesen werden, d.h. die Vergabe einer neuen Slaveadresse erfolgt immer in zwei Schritten. Erst wird die alte Adresse eines AS-I-Slaves gelöscht und dann wird dem AS-I-Slave mit der Adresse 0 die neue Adresse zugewiesen.

Als Abschluß dieses kleinen Überblicks wird auf der Folgeseite in Bild 2-8 eine mögliche Inbetriebnahme demonstriert. Die übergeordnete Steuerungsapplikation meldet sich zuerst mit **'asi_init()'** beim AS-I-Treiber an und nach Beendigung der Kommunikation mit **'asi_shut()'** wieder ab. Die Aufrufsequenz allerdings hat nur beispielhaften Charakter.

☞ **Empfehlungen zum Einsatz**

Für einen ersten Einstieg in die AS-I-Thematik und in die Handhabung der AS-I-Library können die Aufrufe 'asi_init()', 'asi_read_idi()', 'asi_write_odi()' und 'asi_shut()' ausreichen. Mit diesen Aufrufen ist ein ‘Normalbetrieb’ durchführbar.

Für weitergehende Implementierungen stehen die AS-I-Aufrufe entsprechend der AS-I-Master-Spezifikation (/1/) einschließlich administrativer Aufträge wie 'asi_device_info()', 'asi_dia()', 'asi_dia_reset()' etc. zur Verfügung.

Zusätzliche Informationen zu den AS-I-Funktionen finden Sie in /1/.

☞ **Demoprogramm**

Zusätzlich steht Ihnen im Verzeichnis '<lw:>\sinec\cp2413.dw\samples' ein Demoprogramm zur Verfügung.

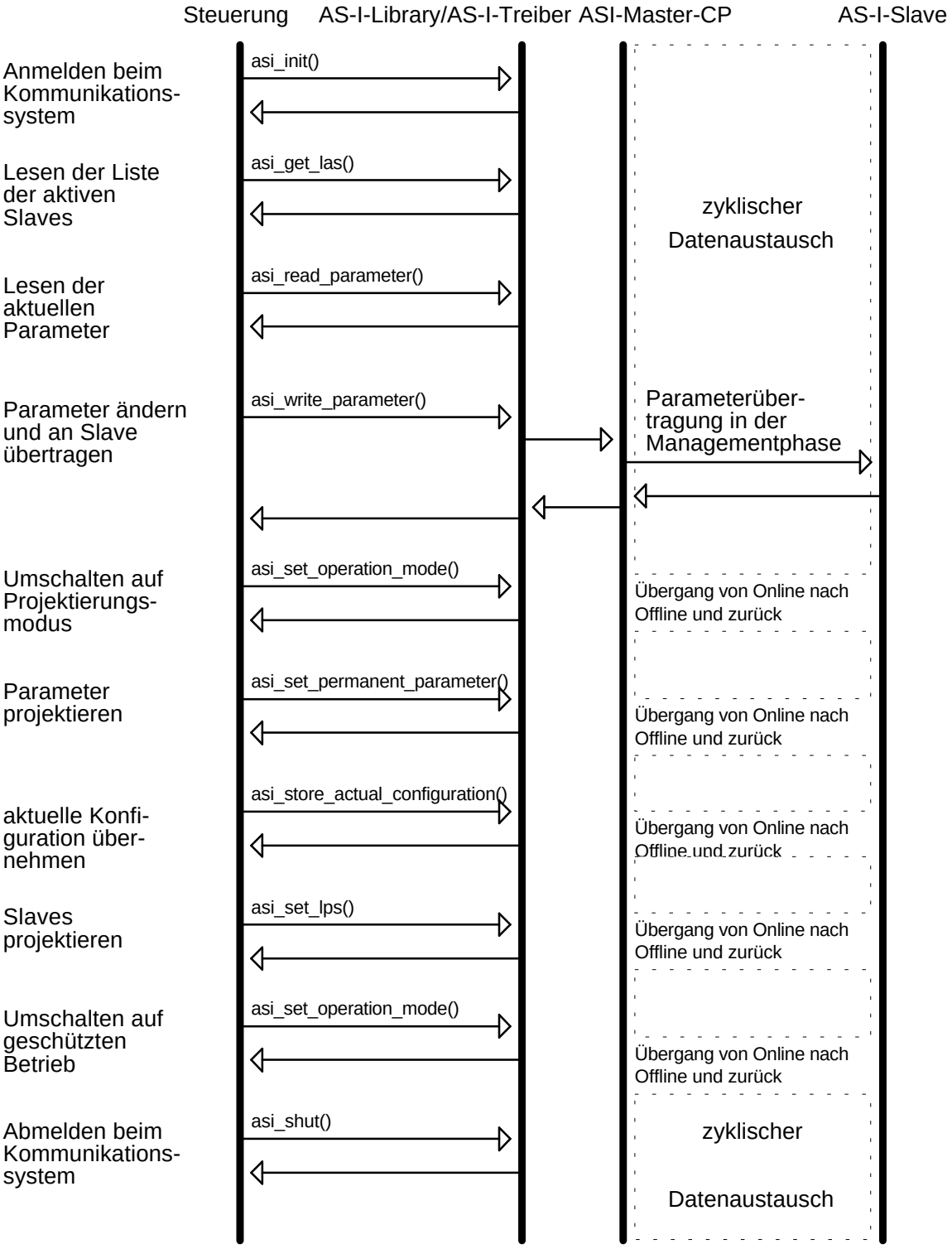


Bild 2-8 Inbetriebnahme (beispielhaft)

Die Kommunikation eines Anwenderprozesses mit den Slaves erfolgt über die einzelnen Bibliotheksfunktionen der AS-I-Library, die in den folgenden Unterkapiteln beschrieben sind.

Der fehlerhafte Ablauf eines Auftrages wird durch einen Returnwert ungleich 'ASI_OK_RESPONSE' gekennzeichnet. Die Returnwerte selbst und ihre Bedeutung sind in Kapitel 2.10.4 aufgeführt.

2.5.2 Funktionsübersicht

Die nachfolgende Tabelle Tabelle 2-2 gibt einen Überblick über die bei "AS-I-2413 MS-DOS/Windows" verfügbaren Funktionen entsprechend der Klasse M1 (siehe auch Anhang PICS).

Nr.	Funktion oder Aufruf an der Host-Schnittstelle (Symbolische Darstellung)	M1	Aufruf der AS-I-Library / Kapitel
1	Image, Status = Read_IDI()	X	asi_read_idi() / 2.5.2.3
2	Status = Write_ODI(Image)	X	asi_write_odi() / 2.5.2.4
3	Status = Set_Permanent_Parameter(Addr, Param)	X	asi_set_permanent_parameter() / 2.5.2.5
4	Param, Status = Get_Permanent_Parameter(Addr)	X	asi_get_permanent_parameter() / 2.5.2.6
5	Status, GParam = Write_Parameter(Addr, Param)	X	asi_write_parameter() / 2.5.2.7
6	Status, Param = Read_Parameter(Addr)	X	asi_read_parameter() / 2.5.2.8
7	Status = Store_Actual_Parameters()	X	asi_store_actual_parameters() / 2.5.2.9
8	Status = Set_Permanent_Configuration(Addr, Config)	X	asi_set_permanent_configuration() / 2.5.2.10
9	Status, Config = Get_Permanent_Configuration(Addr)	X	asi_get_permanent_configuration() / 2.5.2.11
10	Status = Store_Actual_Configuration()	X	asi_store_actual_configration() / 2.5.2.12
11	Status, Config = Read_Actual_Configuration(Addr)	X	asi_read_actual_configuration() / 2.5.2.13
12	Status = Set_LPS(List31)	X	asi_set_lps() / 2.5.2.14
13	Status, List31 = Get_LPS()	X	asi_get_lps() / 2.5.2.15
14	Status, List31 = Get_LAS()	X	asi_get_las() / 2.5.2.16
15	Status, List32 = Get_LDS()	X	asi_get_lds() / 2.5.2.17
16.0	Status = Get_Flags()	X	asi_get_flags() / 2.5.2.18
16.1	Status, Flag = Get_Flag_Config_OK()	X	asi_get_flags() / 2.5.2.18
16.2	Status, Flag = Get_Flag_LDS.0()	X	asi_get_flags() / 2.5.2.18

Nr.	Funktion oder Aufruf an der Host-Schnittstelle (Symbolische Darstellung)	M1	Aufruf der ASI-Library / Kapitel
16.3	Status, Flag = Get_Flag_Auto_Address_Assign()	X	asi_get_flags() / 2.5.2.18
16.4	Status, Flag = Get_Flag_Auto_Prog_Available()	X	asi_get_flags() / 2.5.2.18
16.5	Status, Flag = Get_Flag_Configuration_Active()	X	asi_get_flags() / 2.5.2.18
16.6	Status, Flag = Get_Flag_Normal_Operation_Active()	X	asi_get_flags() / 2.5.2.18
16.7	Status, Flag = Get_Flag_APF()	X	asi_get_flags() / 2.5.2.18
16.8	Status, Flag = Get_Flag_Offline_Ready()	X	asi_get_flags() / 2.5.2.18
17	Status = Set_Operation_Mode(Mode)	X	asi_set_operation_mode() / 2.5.2.19
18	Status = Set_Offline_Mode(Mode)	X	asi_set_offline_mode() / 2.5.2.20
20	Status = Change_Slave_Address(Addr1, Addr2)	X	asi_change_slave_address() / 2.5.2.21

Tabelle 2-2 Funktionen der AS-I-Library entsprechend der Klasse M1

Die Tabelle 2-3 faßt die Funktionen zusammen, die zusätzlich in der AS-I-Library implementiert sind.

	Funktion oder Aufruf an der Host-Schnittstelle (Symbolische Darstellung)	Aufruf der AS-I-Library / Kapitel
	Kommandoaufrufe	
	Status, Resp = Cmd_Reset_ASI_Slave(Addr, RESET)	asi_execute_command() / 2.5.2.23
	Status, Resp = Cmd_Read_IO_Configuration(Addr, CONF)	asi_execute_command() / 2.5.2.23
	Status, Resp = Cmd_Read_Identification_Code(Addr, IDCOD)	asi_execute_command() / 2.5.2.23
	Status, Resp = Cmd_Read_Status(Addr, STAT)	asi_execute_command() / 2.5.2.23
	Status, Resp = Cmd_Read_Reset_Status(Addr, STATRES)	asi_execute_command() / 2.5.2.23
	Status, Resp = Cmd_Write_Read_Data(Addr, DATA)	asi_execute_command() / 2.5.2.23
	Status, Resp = Cmd_Write_Read_Parameter(Addr, PARA)	asi_execute_command() / 2.5.2.23

	Administrative und sonstige Aufrufe	
	Resp = Initialize	asi_init() / 2.5.2.1
	Resp = Shut	asi_shut() / 2.5.2.2
	Resp = Reset	asi_reset() / 2.5.2.22
	Resp = Version_Read	asi_version() / 2.5.2.24
	Resp = Diagnosis_Read	asi_dia() / 2.5.2.25
	Resp = Diagnosis_Reset	asi_dia_reset() / 2.5.2.26
	Resp = Device_Info_Read	asi_device_info() / 2.5.2.27
	Resp = Board_Configuration_Read	asi_cfg_in() / 2.5.2.28
	Resp = Board_Configuration_Write	asi_cfg_out() / 2.5.2.29
	Resp = Read_All	asi_in_all() / 2.5.2.30
	Resp = Watchdog	asi_wd() / 2.5.2.31

Tabelle 2-3 zusätzliche Funktionen der AS-I-Library

2.5.2.1 asi_init

Bedeutung

Mit diesem Aufruf meldet sich eine Anwenderapplikation bei der AS-I-Library und beim AS-I-Treiber zur Kommunikation über eine ausgewählte AS-I-Master-Anschaltung an. Die Prozedur liefert als Rückgabewert einen Handle, der die AS-I-Master-Anschaltung identifiziert und bei weiteren Aufrufen benutzt werden muß.

Deklaration der Funktion:

```
void  *asi_init(dev_name,error)
int8  *dev_name;
ord16 *error;
```

Beschreibung der Parameter:

dev_name	Der Parameter gibt den Namen des CPs vor (z.B. "CP_ASI1:"). Er muß mit einem bei der Inbetriebnahme benutzten Namen für die einzelnen AS-I-CPs identisch sein.
error	Dieser Parameter enthält nach Funktionsbeendigung die Fehlerursache. Er ist nur relevant, wenn ein Fehler erkannt und der NULL-Pointer zurückgegeben wurde.

Rückgabewerte:

!=NULL	Handle, das für die weitere Kommunikation über die gewählte Anschaltung benutzt werden muß.
NULL	Fehlerfall (im Parameter 'error' ist die Fehlerursache enthalten).

Beispiel:

```
void     *handle;
ord16    error;
handle = asi_init((int8 *)"CP_ASI1:",(ord16 *)&error);
```

2.5.2.2 asi_shut

Bedeutung

Mit diesem Aufruf meldet sich der Anwender beim AS-I-Treiber ab. Sämtliche Ressourcen des aufrufenden Prozesses, die dem übergebenen Handle zugeordnet wurden, werden in den Anfangszustand gebracht.

Bei Erfolg ist das Handle nicht mehr gültig und darf nicht weiter benutzt werden.

Deklaration der Funktion:

```
ord16 asi_shut(handle)
void  *handle;
```

Beschreibung der Parameter:

handle	Handle als Rückgabewert des 'asi_init()'-Aufrufs.

Rückgabewerte:

ASI_OK_RESPONSE	Erfolg.
sonst	Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret;
ret = asi_shut(handle);
```

2.5.2.3 asi_read_idi (Eingangsdatenabbild_lesen)

Bedeutung

Mit diesem Aufruf werden die Eingangsdaten der AS-I-Slaves aus dem Eingangsdatenabbild im DPRAM des AS-I-Master-CPs gelesen und in einen vom Anwender bereitgestellten Puffer übertragen.

Deklaration der Funktion:

```
ord16 asi_read_idi(handle,stat_no,buf_len,buffer)
void  *handle;
ord16 stat_no;
ord16 buf_len;
ord8  *buffer;
```

Beschreibung der Parameter:

handle — Handle als Rückgabewert des 'asi_init()-'Aufrufs.

stat_no — Über diesen Parameter kann der Zugriff auf das Eingangsdatenabbild gesteuert werden. Ist Bit 7 gesetzt, so werden die Eingangsdaten sämtlicher Slaves ausgelesen (Eingangsdatenliste), im anderen Fall nur die Eingangsdaten desjenigen Slaves, dessen Adresse in den unteren fünf Bits kodiert ist.

Parameter 'stat_no':

Bit 15-8	Bit 7	Bit 6-5	Bit 4-0
0	0	0	Slave-Adresse
0	1	0	0

buf_len — Dieser Parameter gibt die Länge des übergebenen Puffers an. Beim Lesen der Eingangsdatenliste muß der Puffer mindestens 32 Byte lang sein (max. 32 Slaves können am Netz adressiert werden). Im anderen Fall reicht ein Byte als Anwenderpuffer aus.

buffer Zeiger auf den vom Anwender bereitgestellten Puffer. Die Daten werden hier wie folgt abgelegt (unter Angabe des Offsets zum Pufferanfang):

einzelner Slave:

Offset	Bit 7-4	Bit 3-0
0	0	Eingangsdaten für Slave

Eingangsdatenliste:

Offset	Bit 7-4	Bit 3-0
0	0	irrelevant
1	0	Eingangsdaten für Slave 1
:	:	:
31	0	Eingangsdaten für Slave 31

Rückgabewerte:

ASI_OK_RESPONSE Erfolg.

sonst Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret;
ord8    buffer[ASI_MAX_NO_SLAVES];
ret = asi_read_idi(handle,ASI_LIST,ASI_MAX_NO_SLAVES,buffer);
```

2.5.2.4 asi_write_odi (Ausgangsdatenabbild_schreiben)

Bedeutung

Mit diesem Aufruf werden die Ausgangsdaten aus einem vom Anwender übergebenen Puffer in das Ausgangsdatenabbild im DPRAM des AS-I-Master-CPs übertragen.

Deklaration der Funktion:

```
ord16 asi_write_odi(handle,stat_no,buf_len,buffer)
void  *handle;
ord16 stat_no;
ord16 buf_len;
ord8  *buffer;
```

Beschreibung der Parameter:

handle — Handle als Rückgabewert des 'asi_init-'Aufrufs.

stat_no — Über diesen Parameter kann der Zugriff auf das Ausgangsdatenabbild gesteuert werden. Ist Bit 7 gesetzt, so werden die Ausgangsdaten sämtlicher Slaves geschrieben (Ausgangsdatenliste), im anderen Fall nur die Ausgangsdaten desjenigen Slaves, dessen Adresse in den unteren fünf Bits kodiert ist.

Parameter 'stat_no':

Bit 15-8	Bit 7	Bit 6-5	Bit 4-0
0	0	0	Slave-Adresse
0	1	0	0

buf_len — Dieser Parameter gibt die Länge des übergebenen Puffers an. Beim Schreiben der Ausgangsdatenliste muß der Puffer mindestens 32 Byte lang sein (max. 32 Slaves können am Netz adressiert werden). Im anderen Fall reicht ein Byte als Anwenderpuffer aus.

buffer Zeiger auf den vom Anwender bereitgestellten Puffer. Die Daten müssen wie folgt abgelegt sein (unter Angabe des Offsets zum Pufferanfang):

einzelner Slave:

Offset	Bit 7-4	Bit 3-0
0	0	Ausgangsdaten für Slave

Ausgangsdatenliste:

Offset	Bit 7-4	Bit 3-0
0	0	irrelevant
1	0	Ausgangsdaten für Slave 1
:	:	:
31	0	Ausgangsdaten für Slave 31

Rückgabewerte:

ASI_OK_RESPONSE Erfolg.

sonst Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret;
ord8    buffer[ASI_MAX_NO_SLAVES];
ret = asi_write_odi(handle,ASI_LIST,ASI_MAX_NO_SLAVES,buffer);
```

2.5.2.5 asi_set_permanent_parameter (Parameterwert_projektieren)

Bedeutung

Mit diesem Aufruf werden Parameterdaten für die AS-I-Slaves projektiert. Die Daten werden nichtflüchtig auf der AS-I-Master-Anschaltung gespeichert und ins Abbild der Projektierungsparameter im DPRAM geschrieben. Das Abspeichern der Daten dauert bis zu 640ms.

Deklaration der Funktion:

```
ord16 asi_set_permanent_parameter(handle,stat_no,buf_len,buffer)
void  *handle;
ord16 stat_no;
ord16 buf_len;
ord8  *buffer;
```

Beschreibung der Parameter:

handle — Handle als Rückgabewert des 'asi_init()-'Aufrufs.

stat_no — Über diesen Parameter kann der Zugriff auf die projektierten Parameter gesteuert werden. Ist Bit 7 gesetzt, so werden die Parameter sämtlicher Slaves geschrieben (Parameterliste), im anderen Fall nur die Parameter desjenigen Slaves, dessen Adresse in den unteren fünf Bits kodiert ist.

Parameter 'stat_no':

Bit 15-8	Bit 7	Bit 6-5	Bit 4-0
0	0	0	Slave-Adresse
0	1	0	0

buf_len — Dieser Parameter gibt die Länge des übergebenen Puffers an. Beim Schreiben der Parameterliste muß der Puffer mindestens 32 Byte lang sein (max. 32 Slaves können am Netz adressiert werden). Im anderen Fall reicht ein Byte als Anwenderpuffer aus.

buffer Zeiger auf den vom Anwender bereitgestellten Puffer. Die Daten müssen wie folgt abgelegt sein (unter Angabe des Offsets zum Pufferanfang):

einzelner Slave:

Offset	Bit 7-4	Bit 3-0
0	0	Parameter für Slave

Parameterliste:

Offset	Bit 7-4	Bit 3-0
0	0	irrelevant
1	0	Parameter für Slave 1
:	:	:
31	0	Parameter für Slave 31

Rückgabewerte:

ASI_OK_RESPONSE Erfolg.

sonst Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret;
ord8    buffer[ASI_MAX_NO_SLAVES];
ret = asi_set_permanent_parameter(handle,ASI_LIST,ASI_MAX_NO_SLAVES,buffer);
```

2.5.2.6 asi_get_permanent_parameter (Projektierten_Parameterwert_lesen)

Bedeutung

Mit diesem Aufruf werden die projektierten Parameterdaten der AS-I-Slaves aus dem Abbild der Projektierungsparameter im DPRAM der AS-I-Master-Anschaltung gelesen und in einen vom Anwender bereitgestellten Puffer übertragen.

Deklaration der Funktion:

```
ord16 asi_get_permanent_parameter(handle,stat_no,buf_len,buffer)
void  *handle;
ord16 stat_no;
ord16 buf_len;
ord8  *buffer;
```

Beschreibung der Parameter:

handle — Handle als Rückgabewert des 'asi_init()-'Aufrufs.

stat_no — Über diesen Parameter kann der Zugriff auf die projektierten Parameter gesteuert werden. Ist Bit 7 gesetzt, so werden die Parameter sämtlicher Slaves gelesen (Parameterliste), im anderen Fall nur die Parameter desjenigen Slaves, dessen Adresse in den unteren fünf Bits kodiert ist.

Parameter 'stat_no':

Bit 15-8	Bit 7	Bit 6-5	Bit 4-0
0	0	0	Slave-Adresse
0	1	0	0

buf_len — Dieser Parameter gibt die Länge des übergebenen Puffers an. Beim Lesen der Parameterliste muß der Puffer mindestens 32 Byte lang sein (max. 32 Slaves können am Netz adressiert werden). Im anderen Fall reicht ein Byte als Anwenderpuffer aus.

buffer Zeiger auf den vom Anwender bereitgestellten Puffer. Die Daten werden wie folgt abgelegt (unter Angabe des Offsets zum Pufferanfang):

einzelner Slave:

Offset	Bit 7-4	Bit 3-0
0	0	Parameter für Slave

Parameterliste:

Offset	Bit 7-4	Bit 3-0
0	0	irrelevant
1	0	Parameter für Slave 1
:	:	:
31	0	Parameter für Slave 31

Rückgabewerte:

ASI_OK_RESPONSE Erfolg.

sonst Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret;
ord8    buffer[ASI_MAX_NO_SLAVES];
ret = asi_get_permanent_parameter(handle,ASI_LIST,ASI_MAX_NO_SLAVES,buffer);
```

2.5.2.7 asi_write_parameter (Parameterwert_schreiben)

Bedeutung

Mit diesem Aufruf werden Parameterdaten für die im Segment befindlichen AS-I-Slaves geschrieben und die Einstellungen der Slaves zur Laufzeit geändert. Der Slave übermittelt in der Antwort die aktuellen Parameterwerte, die von den gerade geschriebenen Werten gemäß der AS-I-Master-Spezifikation (/2/) abweichen können. Diese Daten werden im Parameterdatenabbild im DPRAM der AS-I-Master-Anschaltung gehalten und in den beim Aufruf bereitgestellten Anwenderpuffer kopiert. Der Auftrag dauert bis zu zwei Zyklen (ca. 10ms) je angesprochenen Slave.

Deklaration der Funktion:

```
ord16 asi_write_parameter(handle,stat_no,buf_len,buffer)
void  *handle;
ord16 stat_no;
ord16 buf_len;
ord8  *buffer;
```

Beschreibung der Parameter:

handle — Handle als Rückgabewert des 'asi_init()-'Aufrufs.

stat_no — Über diesen Parameter kann der Zugriff auf die zu schreibenden Parameter gesteuert werden. Ist Bit 7 gesetzt, so werden die Parameter sämtlicher Slaves geschrieben (Parameterliste), im anderen Fall nur die Parameter desjenigen Slaves, dessen Adresse in den unteren fünf Bits kodiert ist.

Parameter 'stat_no':

Bit 15-8	Bit 7	Bit 6-5	Bit 4-0
0	0	0	Slave-Adresse
0	1	0	0

buf_len — Dieser Parameter gibt die Länge des übergebenen Puffers an. Beim Schreiben der Parameterliste muß der Puffer mindestens 32 Byte lang sein (max. 32 Slaves können am Netz adressiert werden). Im anderen Fall reicht ein Byte als Anwenderpuffer aus.

buffer Zeiger auf den vom Anwender bereitgestellten Puffer. Die Daten müssen wie folgt abgelegt sein bzw. werden in der Rückantwort wie folgt abgelegt (unter Angabe des Offsets zum Pufferanfang):

einzelner Slave:

Offset	Bit 7-4	Bit 3-0
0	0	Parameter für Slave

Parameterliste:

Offset	Bit 7-4	Bit 3-0
0	0	irrelevant
1	0	Parameter für Slave 1
:	:	:
31	0	Parameter für Slave 31

Rückgabewerte:

ASI_OK_RESPONSE Erfolg.

sonst Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret;
ord8    buffer[ASI_MAX_NO_SLAVES];
ret = asi_write_parameter(handle,ASI_LIST,ASI_MAX_NO_SLAVES,buffer);
```

2.5.2.8 asi_read_parameter (Parameterwert_lesen)

Bedeutung

Mit diesem Aufruf werden die aktuellen Parameterdaten der AS-I-Slaves aus dem Parameterdatenabbild der AS-I-Master-Anschaltung gelesen und in einen vom Anwender bereitgestellten Puffer übertragen.

Deklaration der Funktion:

```
ord16 asi_read_parameter(handle,stat_no,buf_len,buffer)
void  *handle;
ord16 stat_no;
ord16 buf_len;
ord8  *buffer;
```

Beschreibung der Parameter:

handle — Handle als Rückgabewert des 'asi_init()-'Aufrufs.

stat_no — Über diesen Parameter kann der Zugriff auf die aktuellen Daten gesteuert werden. Ist Bit 7 gesetzt, so werden die Parameter sämtlicher Slaves gelesen (Parameterliste), im anderen Fall nur die Parameter desjenigen Slaves, dessen Adresse in den unteren fünf Bits kodiert ist.

Parameter 'stat_no':

Bit 15-8	Bit 7	Bit 6-5	Bit 4-0
0	0	0	Slave-Adresse
0	1	0	0

buf_len — Dieser Parameter gibt die Länge des übergebenen Puffers an. Beim Lesen der Parameterliste muß der Puffer mindestens 32 Byte lang sein (max. 32 Slaves können am Netz adressiert werden). Im anderen Fall reicht ein Byte als Anwenderpuffer aus.

buffer Zeiger auf den vom Anwender bereitgestellten Puffer. Die Daten werden wie folgt abgelegt (unter Angabe des Offsets zum Pufferanfang):

einzelner Slave:

Offset	Bit 7-4	Bit 3-0
0	0	Parameter für Slave

Parameterliste:

Offset	Bit 7-4	Bit 3-0
0	0	irrelevant
1	0	Parameter für Slave 1
:	:	:
31	0	Parameter für Slave 31

Rückgabewerte:

ASI_OK_RESPONSE Erfolg.

sonst Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret;
ord8    buffer[ASI_MAX_NO_SLAVES];
ret = asi_read_parameter(handle,ASI_LIST,ASI_MAX_NO_SLAVES,buffer);
```

2.5.2.9 asi_store_actual_parameters (Ist_Parameterwerte_projektieren)

Bedeutung

Mit diesem Aufruf werden die projektierten Parameter mit den aktuellen Parametern überschrieben, d.h. es erfolgt eine Projektierung der Parameter mit den aktuellen Werten der AS-I-Slaves. Das nichtflüchtige Speichern der Daten dauert bis zu 640ms.

Deklaration der Funktion:

```
ord16   asi_store_actual_parameters(handle);
void    *handle;
```

Beschreibung der Parameter:

handle	Handle als Rückgabewert des 'asi_init()-'Aufrufs.

Rückgabewerte:

ASI_OK_RESPONSE	Erfolg.
sonst	Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret;
ret = asi_store_actual_parameters(handle);
```

2.5.2.10 asi_set_permanent_configuration (Konfigurationsdaten_Projektieren)

Bedeutung

Mit diesem Aufruf werden die E/A-Konfigurationsdaten und die ID-Codes für die AS-I-Slaves projektiert. Die Daten werden nichtflüchtig auf der AS-I-Master-Anschaltung gespeichert und ins Abbild der Konfigurationsdaten geschrieben. Der Auftrag dauert bis zu 640ms.

Deklaration der Funktion:

```
ord16 asi_set_permanent_configuration(handle,stat_no,buf_len,buffer)
void  *handle;
ord16 stat_no;
ord16 buf_len;
ord8  *buffer;
```

Beschreibung der Parameter:

handle — Handle als Rückgabewert des 'asi_init()-'Aufrufs.

stat_no — Über diesen Parameter kann der Zugriff auf die zu projektierenden Konfigurationsdaten gesteuert werden. Ist Bit 7 gesetzt, so werden die Daten sämtlicher Slaves geschrieben (Konfigurationsliste), im anderen Fall nur die Daten desjenigen Slaves, dessen Adresse in den unteren fünf Bits kodiert ist.

Parameter 'stat_no':

Bit 15-8	Bit 7	Bit 6-5	Bit 4-0
0	0	0	Slave-Adresse
0	1	0	0

buf_len — Dieser Parameter gibt die Länge des übergebenen Puffers an. Beim Schreiben der Konfigurationsliste muß der Puffer mindestens 32 Byte lang sein (für die E/A-Konfigurationsdaten und ID-Codes von max. 32 am Netz adressierbaren Slaves). Im anderen Fall reicht ein Byte als Anwenderpuffer aus.

buffer Zeiger auf den vom Anwender bereitgestellten Puffer. Die Daten müssen wie folgt abgelegt sein (unter Angabe des Offsets zum Pufferanfang):

einzelner Slave:

Offset	Bit 7-4	Bit 3-0
0	ID-Code für Slave	E/A-Konfig. Parameter für Slave

Konfigurationsliste:

Offset	Bit 7-4	Bit 3-0
0	ID-Code für Slave 0	E/A-Konfig. Parameter für Slave 0
1	ID-Code für Slave 1	E/A-Konfig. Parameter für Slave 1
:	:	:
31	ID-Code für Slave 31	E/A-Konfig. Parameter für Slave 31

Rückgabewerte:

ASI_OK_RESPONSE Erfolg.

sonst Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret;
ord8    buffer[ASI_MAX_NO_SLAVES];
ret = asi_set_permanent_configuration(handle, ASI_LIST, ASI_MAX_NO_SLAVES,buffer);
```

2.5.2.11 asi_get_permanent_configuration (Projektierte_Konfigurationsdaten_Lesen)

Bedeutung

Mit diesem Aufruf werden die projektierten E/A-Konfigurationsdaten und die ID-Codes der AS-I-Slaves aus dem Abbild der Konfigurationsdaten im DPRAM ausgelesen und in einen vom Anwender bereitgestellten Puffer übertragen.

Deklaration der Funktion:

```
ord16 asi_get_permanent_configuration(handle,stat_no,buf_len,buffer)
void  *handle;
ord16 stat_no;
ord16 buf_len;
ord8  *buffer;
```

Beschreibung der Parameter:

handle — Handle als Rückgabewert des 'asi_init()-'Aufrufs.

stat_no — Über diesen Parameter kann der Zugriff auf die projektierten Konfigurationsdaten gesteuert werden. Ist Bit 7 gesetzt, so werden die Daten sämtlicher Slaves gelesen (Konfigurationsliste), im anderen Fall nur die Daten desjenigen Slaves, dessen Adresse in den unteren fünf Bits kodiert ist.

Parameter 'stat_no':

Bit 15-8	Bit 7	Bit 6-5	Bit 4-0
0	0	0	Slave-Adresse
0	1	0	0

buf_len — Dieser Parameter gibt die Länge des übergebenen Puffers an. Beim Lesen der Konfigurationsliste muß der Puffer mindestens 32 Byte lang sein (für die E/A-Konfigurationsdaten und ID-Codes von max. 32 am Netz adressierbaren Slaves). Im anderen Fall reicht ein Byte als Anwenderpuffer aus.

buffer Zeiger auf den vom Anwender bereitgestellten Puffer. Die Daten werden wie folgt abgelegt (unter Angabe des Offsets zum Pufferanfang):

einzelner Slave:

Offset	Bit 7-4	Bit 3-0
0	ID-Code für Slave	E/A-Konfig. Parameter für Slave

Konfigurationsliste:

Offset	Bit 7-4	Bit 3-0
0	irrelevant	irrelevant
1	ID-Code für Slave 1	E/A-Konfig. Parameter für Slave 1
:	:	:
31	ID-Code für Slave 31	E/A-Konfig. Parameter für Slave 31

Rückgabewerte:

ASI_OK_RESPONSE Erfolg.

sonst Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret;
ord8    buffer[ASI_MAX_NO_SLAVES];
ret = asi_get_permanent_configuration(handle, ASI_LIST, ASI_MAX_NO_SLAVES,buffer);
```

2.5.2.12 asi_store_actual_configuration (Ist_Konfigurationsdaten_Projektieren)

Bedeutung

Mit diesem Aufruf werden die am AS-I ermittelten (IST-) E/A-Konfigurationsdaten und (IST-) ID-Codes nichtflüchtig im EEPROM als SOLL-Konfigurationsdaten gespeichert. Ebenso wird die Liste der aktivierten Slaves (LAS) in die Liste der projektierten Slaves (LPS) übernommen. Der Auftrag dauert bis zu 640ms.

☞ **Bei der Durchführung dieses Kommandos wechselt der CP in die Offline-Phase und geht anschließend wieder in den Normalbetrieb über (Neustart des CP).**

Im geschützten Betrieb wird dieser Aufruf **nicht** durchgeführt.

Deklaration der Funktion:

```
ord16 asi_store_actual_configuration(handle);
void  *handle;
```

Beschreibung der Parameter:

handle	Handle als Rückgabewert des 'asi_init()-'Aufrufs.

Rückgabewerte:

ASI_OK_RESPONSE	Erfolg.
sonst	Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret;
ret = asi_store_actual_configuration(handle);
```

2.5.2.13 asi_read_actual_configuration (Ist_Konfigurationsdaten_Lesen)

Mit diesem Aufruf werden die aktuellen E/A-Konfigurationsdaten und die ID-Codes der AS-I-Slaves aus dem Abbild der Konfigurationsdaten im DPRAM ausgelesen und in einen vom Anwender bereitgestellten Puffer übertragen.

Deklaration der Funktion:

```
ord16 asi_read_actual_configuration(handle,stat_no,buf_len_buffer)
void  *handle;
ord16 stat_no;
ord16 buf_len;
ord8  *buffer;
```

Beschreibung der Parameter:

handle — Handle als Rückgabewert des 'asi_init()-'Aufrufs.

stat_no — Über diesen Parameter kann der Zugriff auf die aktuellen Konfigurationsdaten gesteuert werden. Ist Bit 7 gesetzt, so werden die Daten sämtlicher Slaves gelesen (Konfigurationsliste), im anderen Fall nur die Daten desjenigen Slaves, dessen Adresse in den unteren fünf Bits kodiert ist.

Parameter 'stat_no':

Bit 15-8	Bit 7	Bit 6-5	Bit 4-0
0	0	0	Slave-Adresse
0	1	0	0

buf_len — Dieser Parameter gibt die Länge des übergebenen Puffers an. Beim Lesen der Konfigurationsliste muß der Puffer mindestens 32 Byte lang sein (für die E/A-Konfigurationsdaten und ID-Codes von max. 32 am Netz adressierbaren Slaves). Im anderen Fall reicht ein Byte als Anwenderpuffer aus.

buffer Zeiger auf den vom Anwender bereitgestellten Puffer. Die Daten werden wie folgt abgelegt (unter Angabe des Offsets zum Pufferanfang):

einzelner Slave:

Offset	Bit 7-4	Bit 3-0
0	ID-Code für Slave	E/A-Konfig. Parameter für Slave

Konfigurationsliste:

Offset	Bit 7-4	Bit 3-0
0	irrelevant	irrelevant
1	ID-Code für Slave 1	E/A-Konfig. Parameter für Slave 1
:	:	:
31	ID-Code für Slave 31	E/A-Konfig. Parameter für Slave 31

Rückgabewerte:

ASI_OK_RESPONSE Erfolg.

sonst Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret;
ord8    buffer[ASI_MAX_NO_SLAVES];
ret = asi_read_actual_configuration(handle,ASI_LIST,ASI_MAX_NO_SLAVES,buffer);
```

2.5.2.14 asi_set_lps (LPS_Projektieren)

Bedeutung

Mit diesem Aufruf wird die Liste der projektierten Slaves übergeben. Die Daten werden nichtflüchtig auf der AS-I-Master-Anschaltung gespeichert und ins Abbild der LPS im DPRAM geschrieben. Der Auftrag dauert bis zu 70ms.

☞ **Bei der Durchführung dieses Kommandos wechselt der CP in die Offline-Phase und geht anschließend wieder in den Normalbetrieb über (Neustart des CP).**

Im geschützten Betrieb wird dieser Aufruf **nicht** durchgeführt.

Deklaration der Funktion:

```
ord16 asi_set_lps(handle,buf_len,buffer)
void  *handle;
ord16 buf_len;
ord8  *buffer;
```

Beschreibung der Parameter:

handle — Handle als Rückgabewert des 'asi_init()-'Aufrufs.

buf_len — Dieser Parameter gibt die Länge des übergebenen Puffers an. Als Anwenderpuffer werden mindestens 4 Byte zur bitweisen Kodierung der bis zu 32 Slaves benötigt.

buffer — Zeiger auf den vom Anwender bereitgestellten Puffer. Die Daten müssen wie folgt abgelegt sein (unter Angabe des Offsets zum Pufferanfang), wobei das für den Slave x reservierte Bit gesetzt sein muß, wenn der Slave projektiert werden soll:

Liste der projektierten Slaves (LPS):

Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Slave7	Slave6	Slave5	Slave4	Slave3	Slave2	Slave1	0
1	Slave15	Slave14	Slave13	Slave12	Slave11	Slave10	Slave9	Slave8
2	Slave23	Slave22	Slave21	Slave20	Slave19	Slave18	Slave17	Slave16
3	Slave31	Slave30	Slave29	Slave28	Slave27	Slave26	Slave25	Slave24

Rückgabewerte:

ASI_OK_RESPONSE — Erfolg.

sonst — Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret;
ord32   lps;
ret = asi_set_lps(handle,sizeof(ord32),(ord8 *)&lps);
```

2.5.2.15 asi_get_lps (LPS_Lesen)

Bedeutung

Mit diesem Aufruf wird die Liste der projektierten Slaves aus dem Abbild der LPS im DPRAM ausgelesen.

Deklaration der Funktion:

```
ord16 asi_get_lps(handle,buf_len,buffer)
void  *handle;
ord16 buf_len;
ord8  *buffer;
```

Beschreibung der Parameter:

handle — Handle als Rückgabewert des 'asi_init()-'Aufrufs.

buf_len — Dieser Parameter gibt die Länge des übergebenen Puffers an. Als Anwenderpuffer werden mindestens 4 Byte zur bitweisen Kodierung der bis zu 32 Slaves benötigt.

buffer — Zeiger auf den vom Anwender bereitgestellten Puffer. Die Daten werden wie folgt abgelegt (unter Angabe des Offsets zum Pufferanfang), wobei das für den Slave x reservierte Bit gesetzt ist, wenn der Slave zuvor projektiert wurde:

Liste der projektierten Slaves (LPS):

Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Slave7	Slave6	Slave5	Slave4	Slave3	Slave2	Slave1	Slave0
1	Slave15	Slave14	Slave13	Slave12	Slave11	Slave10	Slave9	Slave8
2	Slave23	Slave22	Slave21	Slave20	Slave19	Slave18	Slave17	Slave16
3	Slave31	Slave30	Slave29	Slave28	Slave27	Slave26	Slave25	Slave24

Rückgabewerte:

ASI_OK_RESPONSE — Erfolg.

sonst — Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret;
ord32   lps;
ret = asi_get_lps(handle,sizeof(ord32),(ord8 *)&lps);
```

2.5.2.16 asi_get_las (LAS_Lesen)

Bedeutung

Mit diesem Aufruf wird die Liste der aktiven Slaves aus dem Abbild der LAS im DPRAM ausgelesen.

Deklaration der Funktion:

```
ord16 asi_get_las(handle,buf_len,buffer)
void  *handle;
ord16 buf_len;
ord8  *buffer;
```

Beschreibung der Parameter:

handle — Handle als Rückgabewert des 'asi_init()-'Aufrufs.

buf_len — Dieser Parameter gibt die Länge des übergebenen Puffers an. Als Anwenderpuffer werden mindestens 4 Byte zur bitweisen Kodierung der bis zu 32 Slaves benötigt.

buffer — Zeiger auf den vom Anwender bereitgestellten Puffer. Die Daten werden wie folgt abgelegt (unter Angabe des Offsets zum Pufferanfang), wobei das für den Slave x reservierte Bit gesetzt ist, wenn der Slave aktiviert werden konnte:

Liste der aktiven Slaves (LAS):

Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Slave7	Slave6	Slave5	Slave4	Slave3	Slave2	Slave1	Slave0
1	Slave15	Slave14	Slave13	Slave12	Slave11	Slave10	Slave9	Slave8
2	Slave23	Slave22	Slave21	Slave20	Slave19	Slave18	Slave17	Slave16
3	Slave31	Slave30	Slave29	Slave28	Slave27	Slave26	Slave25	Slave24

Rückgabewerte:

ASI_OK_RESPONSE — Erfolg.

sonst — Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret;
ord32   las;

ret = asi_get_las(handle,sizeof(ord32),(ord8 *)&las);
```

2.5.2.17 asi_get_lds (LDS_Lesen)

Bedeutung

Mit diesem Aufruf wird die Liste der erkannten Slaves aus dem Abbild der LDS im DPRAM ausgelesen.

Deklaration der Funktion:

```
ord16 asi_get_lds(handle,buf_len,buffer)
void  *handle;
ord16 buf_len;
ord8  *buffer;
```

Beschreibung der Parameter:

handle — Handle als Rückgabewert des 'asi_init()-'Aufrufs.

buf_len — Dieser Parameter gibt die Länge des übergebenen Puffers an. Als Anwenderpuffer werden mindestens 4 Byte zur bitweisen Kodierung der bis zu 32 Slaves benötigt.

buffer — Zeiger auf den vom Anwender bereitgestellten Puffer. Die Daten werden wie folgt abgelegt (unter Angabe des Offsets zum Pufferanfang), wobei das für den Slave x reservierte Bit gesetzt ist, wenn der Slave erkannt wurde:

Liste der erkannten Slaves (LDS):

Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Slave7	Slave6	Slave5	Slave4	Slave3	Slave2	Slave1	Slave0
1	Slave15	Slave14	Slave13	Slave12	Slave11	Slave10	Slave9	Slave8
2	Slave23	Slave22	Slave21	Slave20	Slave19	Slave18	Slave17	Slave16
3	Slave31	Slave30	Slave29	Slave28	Slave27	Slave26	Slave25	Slave24

Rückgabewerte:

ASI_OK_RESPONSE — Erfolg.

sonst — Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret;
ord32   lds;

ret = asi_get_lds(handle,sizeof(ord32),(ord8 *)&lds);
```

2.5.2.18 asi_get_flags (Flags_Lesen)

Bedeutung

Mit diesem Aufruf werden die Flags aus dem Abbild im DPRAM der AS-I-Master-Anschaltung ausgelesen.

Deklaration der Funktion:

```
ord16 asi_get_flags(handle,buf_len,buffer)
void  *handle;
ord16 buf_len;
ord8  *buffer;
```

Beschreibung der Parameter:

handle — Handle als Rückgabewert des 'asi_init()-'Aufrufs.

buf_len — Dieser Parameter gibt die Länge des übergebenen Puffers an. Als Anwenderpuffer werden mindestens 2 Byte für die Flags benötigt.

buffer — Zeiger auf den vom Anwender bereitgestellten Puffer. Die Flags werden als Wort im Anwenderpuffer abgelegt, wobei die einzelnen Bits die im folgenden beschriebene Bedeutung haben:

Flag	Bedeutung
ASI_OFFLINE_READY_FLAG	Offline_ready
ASI_APF_FLAG	APF
ASI_NORMAL_OPERATION_ACTIVE_FLAG	Normalbetrieb_aktiv
ASI_CONFIGURATION_ACTIVE_FLAG	Projektierung_aktiv
ASI_AUTO_PROG_AVAILABLE_FLAG	Auto_prog_available
ASI_AUTO_ADDRESS_ASSIGN_FLAG	Auto_prog
ASI_LDS_0_FLAG	LDS.0
ASI_CONFIG_OK_FLAG	Konfig_Ok
ASI_OFFLINE_FLAG	Offline
ASI_EEP_OK_FLAG	EEPROM_Ok
ASI_WATCHDOG_FLAG	Watchdog

Das EEPROM_Ok-Flag bleibt gesetzt, solange kein Fehler während des Speichervorganges beim Projektieren auftritt.

Das Watchdog-Flag wird gesetzt, wenn der Watchdog die AS-I-Master-Baugruppe zurückgesetzt hat (siehe auch Funktion 'asi_wd()' zwecks Aktivitätsüberwachung).

Die Bedeutung der restlichen Flags kann der AS-I-Master-Spezifikation (/2/) entnommen werden.

Rückgabewerte:

ASI_OK_RESPONSE	Erfolg.
sonst	Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret,flag;

ret = asi_get_flags(handle,sizeof(ord16),(ord8 *)&flag);
```

2.5.2.19 asi_set_operation_mode (Projektierungsmodus_Setzen)

Bedeutung

Dieser Aufruf schaltet in einen der beiden erlaubten Betriebsmodi.

Beim Wechsel vom Projektierungsmodus in den geschützten Betrieb erfolgt ein Neustart des CP (Übergang in die Offlinephase und anschließendes Umschalten in den Onlinebetrieb).

Beim geschützten Betriebsmodus werden nur Slaves aktiviert, die in der LPS vermerkt sind und deren Soll- und Ist-Konfiguration übereinstimmen, d.h. wenn die E/A-Konfiguration und die Identifikationscodes der in der LDS enthaltenen Slaves mit den projektierten Werten identisch sind.

Im Projektierungsmodus werden alle erkannten Slaves auch bei Unterschieden in der Soll- und Ist-Konfiguration aktiviert.

Deklaration der Funktion:

```
ord16 asi_set_operation_mode(handle,value)
void  *handle;
ord16 value;
```

Beschreibung der Parameter:

handle — Handle als Rückgabewert des 'asi_init()-'Aufrufs.

value — Der Wert dieses Parameters gibt den einzustellenden Betriebsmodus an:

Wertebereich:

Wert	Bedeutung
ASI_SET_PROTECTED_MODE	Umschalten in den geschützten Betrieb
ASI_SET_CONFIGURATION_MODE	Umschalten in den Projektierungsmodus

Rückgabewerte:

ASI_OK_RESPONSE — Erfolg.

sonst — Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret;

ret = asi_set_operation_mode(handle,ASI_SET_CONFIGURATION_MODE);
```

2.5.2.20 asi_set_offline_mode (Offlinemodus_Setzen)

Bedeutung

Deser Aufruf schaltet zwischen dem Online- und dem Offline-Betrieb um und dauert bis zu 2 AS-I-Zyklen.

Der Online-Betrieb stellt den Normalfall der AS-I-Master-Anschaltung dar. Hier werden zyklisch die folgenden Aufträge abgearbeitet:

- In der sogenannten Datenaustauschphase werden für alle Slaves der LAS die Felder der Ausgangsdaten an die Slaveausgänge übertragen. Die angesprochenen Slaves übermitteln bei fehlerfreier Übertragung dem Master die Werte der Slaveeingänge.
- Daran schließt sich die Aufnahmephase an, in der nach den am AS-I vorhandenen Slaves gesucht und neu hinzugekommene Slaves in die LDS bzw. LAS übernommen werden.
- In der Managementphase werden vom Anwender durchgereichte Aufträge wie das Schreiben von Parametern ausgeführt.

Im Offline-Betrieb werden lediglich Aufträge vom Anwender bearbeitet.

Deklaration der Funktion:

```
ord16 asi_set_offline_mode(handle,value)
void  *handle;
ord16 value;
```

Beschreibung der Parameter:

handle — Handle als Rückgabewert des 'asi_init()-'Aufrufs.

value — Der Wert dieses Parameters gibt den einzustellenden Modus an:

Wertebereich:

Wert	Bedeutung
ASI_SET_ONLINE	Umschalten in den Online-Betrieb
ASI_SET_OFFLINE	Umschalten in den Offline-Betrieb

Rückgabewerte:

ASI_OK_RESPONSE — Erfolg.

sonst — Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret;

ret = asi_set_offline_mode(handle,ASI_SET_ONLINE);
```

2.5.2.21 asi_change_slave_address (Betriebsadresse_Ändern)

Bedeutung

Mit diesem Aufruf kann die Betriebsadresse eines Slaves geändert werden

Dieser Aufruf wird vorwiegend verwendet, um einen neuen AS-I-Slave mit der Default-Adresse 0 dem SINEC S1-System hinzuzufügen. In diesem Fall erfolgt eine Adreßänderung von Slave-Adresse-alt=0 auf Slave-Adresse-neu.

Die Änderung erfolgt unter folgenden Bedingungen:

1. Es ist ein Slave mit *Slave-Adresse-alt* vorhanden.
2. Ist die alte Slaveadresse ungleich 0, dann darf nicht gleichzeitig ein Slave mit Adresse 0 angeschlossen sein.
3. Die *Slave-Adresse-neu* muß einen gültigen Wert haben.
4. Ein Slave mit *Slave-Adresse-neu* darf nicht existieren.

Die Statusanzeige signalisiert bei der Rückgabe, ob das Kommando fehlerfrei durchgeführt werden konnte.

Anmerkung: Beim Ändern der Betriebsadresse wird der Slave nicht zurückgesetzt, d.h., daß die Daten erhalten bleiben, bis auf der neuen Adresse neue Daten kommen.

Deklaration der Funktion:

```
ord16 asi_change_slave_address(handle,old_address,new_address)
void  *handle;
ord16 old_address;
ord16 new_address;
```

Beschreibung der Parameter:

handle — Handle als Rückgabewert des 'asi_init()-'Aufrufs.

old_address — Hier wird die Adresse des Slaves angegeben, dessen Adresse geändert werden soll.

new_address — Hier wird die neue Slaveadresse übergeben.

Slaveadresse:

Bit 15-5	Bit 4-0
0	5-Bit-Adresse

Rückgabewerte:

ASI_OK_RESPONSE — Erfolg.

sonst — Fehlerfall.

Beispiel:

```
void     *handle;
ord16    ret,old_address,new_address;

ret = asi_change_slave_address(handle,old_address,new_address);
```

2.5.2.22 asi_reset

Bedeutung

Mit diesem Aufruf werden sämtliche Slaves zurückgesetzt, indem

- ➢ in den Offline-Betrieb geschaltet wird, wenn diese Betriebsart nicht zuvor schon aktiv war,
- ➢ danach wieder in den Online-Betrieb zurückgeschaltet wird, falls diese Betriebsart vor dem Reset-Aufruf eingestellt war.

Deklaration der Funktion:

```
ord16 asi_reset(handle)
void  *handle;
```

Beschreibung der Parameter:

handle	Handle als Rückgabewert des 'asi_init()-'Aufrufs.

Rückgabewerte:

ASI_OK_RESPONSE	Erfolg.
sonst	Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret;

ret = asi_reset(handle);
```

2.5.2.23 asi_execute_command

Bedeutung

Mit diesem Aufruf können direkt Aufrufe an die AS-I-Slaves abgesetzt werden. Dabei wird nach den folgenden Funktionsgruppen unterschieden:

- ➢ Datenaufruf,
- ➢ Parameteraufruf,
- ➢ Konfigurationsaufruf.

Beim Konfigurationsaufruf wird detaillierter unterschieden zwischen:

- ➢ Reset Slave,
- ➢ E/A-Konfiguration lesen,
- ➢ ID-Code lesen,
- ➢ Status lesen und
- ➢ Status lesen & löschen.

Der Auftrag dauert bis zu zwei AS-I-Zyklen (10ms).

Deklaration der Funktion:

```
ord16 asi_execute_command(handle,cmd,address,data)
void  *handle;
ord16 cmd;
ord16 address;
ord8  *data;
```

Beschreibung der Parameter:

handle — Handle als Rückgabewert des 'asi_init()-'Aufrufs.

cmd — Hier wird eine Kennung für den Aufruf übergeben.

Wertebereich für 'cmd':

Wert	Bedeutung
ASI_CMD_DATA	Datenaufruf
ASI_CMD_PARA	Parametrieraufruf
ASI_CMD_RESET	Reset Slave
ASI_CMD_READ_CONFIG	E/A-Konfiguration lesen
ASI_CMD_READ_ID	ID-Code lesen
ASI_CMD_READ_STATE	Status lesen
ASI_CMD_DEL_STATE	Status lesen und löschen

address Mit diesem Parameter wird der Slave selektiert, der mit dem Aufruf angesprochen werden soll.

Slaveadresse:

Bit 15-5	Bit 4-0
0	5-Bit-Adresse

data Bei einem Daten- oder Parameteraufruf werden hier die Bitmuster für die Ausgänge bzw. die Parameter des selektierten Slaves an die Library übergeben. Bei allen Aufträgen trägt die AS-I-Library in dieses Feld die Slaveantwort ein.

Daten / Parameter bzw. Slaveantwort:

Bit 7-4	Bit 3-0
0	4-Bit-Daten / Parameter bzw. Slaveantwort

Rückgabewerte:

ASI_OK_RESPONSE Erfolg.

sonst Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret,address;
ord8    data;

ret = asi_execute_command(handle,ASI_CMD_DATA,address,&data);
```

2.5.2.24 asi_version

Bedeutung

Mit diesem Aufruf wird die Versionskennung der AS-I-Master-Software ausgelesen und an den Anwender übergeben.

Deklaration der Funktion:

```
ord16 asi_version(handle,buf_len,buffer)
void  *handle;
ord16 buf_len;
ord8  *buffer;
```

Beschreibung der Parameter:

handle — Handle als Rückgabewert des 'asi_init()-'Aufrufs.

buf_len — Dieser Parameter gibt die Länge des übergebenen Puffers an, der mindestens die Größe einer Variablen der nachfolgenden Struktur haben muß:

```
typedef struct asi_version_s
{
      ord8  serial_number[ASI_SER_NO_LEN];
      ord8  vendor[ASI_VENDOR_LEN];
      ord8  model[ASI_MODEL_LEN];
      ord8  revision[ASI_REVISION_LEN];
      ord16 ser_no_len;
      ord16 vendor_len;
      ord16 model_len;
      ord16 revision_len;
}     asi_version_t;
```

buffer — Zeiger auf den vom Anwender bereitgestellten Puffer. Die Daten werden entsprechend der obigen Struktur abgelegt; das sind die Serialisierungsnummer, der Name des Herstellers, der Gerätetyp/Model und die Version der AS-I-Master-Software. Die Namen sind alle mit '\0' abgeschlossen, zusätzlich ist die Länge der Namen in den darauffolgenden Worten abgelegt.

Rückgabewerte:

ASI_OK_RESPONSE — Erfolg.

sonst — Fehlerfall.

Beispiel:

```
void          *handle;
ord16         ret;
asi_version_t asi_version_var;

ret = asi_version(handle,sizeof(asi_version_t),(ord8 *)&asi_version_var);
```

2.5.2.25 asi_dia

Bedeutung

Mit diesem Aufruf werden alle Diagnoseinformationen von der AS-I-Master-Anschaltung ausgelesen. Als Diagnoseinformation steht die Anzahl der Fehler zur Verfügung, die bei der Kommunikation mit einem Slave aufgetreten sind.

Der Wert des Fehlerzählers wird jeweils in folgenden Fällen erhöht:

- ➢ Wenn ein Slave in der Datenaustauschphase trotz Wiederholung des Mastertelegrammes nicht korrekt antwortet (d.h. fehlendes oder falsches Slavetelegramm).
- ➢ Bei nicht aktivierten Slaves wird der Fehlerzähler dann erhöht, wenn ein Slave bereits erkannt wurde (d.h. in der LDS vorhanden ist) und anschließend nicht korrekt antwortet.

Der Fehlerzähler wird bis zum Maximalwert von 255 erhöht und bleibt bis zum Löschen auf diesem Wert.

Deklaration der Funktion:

```
ord16 asi_dia(handle,stat_no,buf_len,buffer)
void  *handle;
ord16 stat_no;
ord16 buf_len;
ord8  *buffer;
```

Beschreibung der Parameter:

handle — Handle als Rückgabewert des 'asi_init()-'Aufrufs.

stat_no — Über diesen Parameter kann der Zugriff auf die aktuellen Daten gesteuert werden. Ist Bit 7 gesetzt, so werden die Fehler sämtlicher Slaves gelesen (Fehlerliste), im anderen Fall nur die Fehleranzahl desjenigen Slaves, dessen Adresse in den unteren fünf Bits kodiert ist.

Parameter 'stat_no':

Bit 15-8	Bit 7	Bit 6-5	Bit 4-0
0	0	0	Slave-Adresse
0	1	0	0

buf_len — Dieser Parameter gibt die Länge der übergebenen Puffers an. Beim Lesen der Fehlerliste muß der Puffer mindestens 32 Byte lang sein (die Anzahl der Fehler wird je Slave in einem Byte gezählt). Im anderen Fall reicht ein Byte als Anwenderpuffer aus.

buffer Zeiger auf den vom Anwender bereitgestellten Puffer. Die Daten werden wie folgt abgelegt (unter Angabe des Offsets zum Pufferanfang):

einzelner Slave:

Offset	Bit 7-0
0	Anzahl der Fehler für Slave

Fehlerliste:

Offset	Bit 7-0
0	Anzahl der Fehler für Slave 0
:	:
31	Anzahl der Fehler für Slave 31

Rückgabewerte:

ASI_OK_RESPONSE Erfolg.

sonst Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret;
ord8    buffer[ASI_MAX_NO_SLAVES];

ret = asi_dia(handle,ASI_LIST,ASI_MAX_NO_SLAVES,buffer);
```

2.5.2.26 asi_dia_reset

Bedeutung

Mit diesem Aufruf wird die Fehlerliste zurückgesetzt.

Deklaration der Funktion:

```
ord16 asi_dia_reset(handle);
void  *handle;
```

Beschreibung der Parameter:

handle	Handle als Rückgabewert des 'asi_init()-'Aufrufs.

Rückgabewerte:

ASI_OK_RESPONSE	Erfolg.
sonst	Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret;

ret = asi_dia_reset(handle);
```

2.5.2.27 asi_device_info

Bedeutung

Mit diesem Aufruf gibt der Treiber die Anzahl der verfügbaren AS-I-Master-Anschaltungen und die Namen, mit denen diese angesprochen werden können, bekannt. Darüberhinaus werden sämtliche hardwareabhängigen Parameter angezeigt.

Deklaration der Funktion:

```
ord16   asi_device_info(buf_len,buffer)
ord16   buf_len;
ord8    *buffer;
```

Beschreibung der Parameter:

buf_len — Dieser Parameter gibt die Länge des übergebenen Puffers an, der mindestens die Größe einer Variablen der nachfolgenden Struktur haben muß:

```
typedef struct asi_device_info_s
{     ord16       num_boards;
      asi_device_t      device[ASI_MAX_NO_BOARDS];
      }     asi_device_info_t;
mit
typedef struct asi_device_s
{
      ord16 dpram_seg;
      ord16 name_len;
      int8  name[ASI_MAX_DRV_NAME_LEN+2];
      ord16 io_port;
}     asi_device_t;
```

buffer — Zeiger auf den vom Anwender bereitgestellten Puffer. Die Daten werden entsprechend der obigen Struktur abgelegt, das sind die Anzahl der verfügbaren AS-I-Master-Anschaltungen, die Ansprechnamen, die Segmentadressen der DPRAMs sowie die Werte der Basis-Konfigurationsregisters.

Rückgabewerte:

ASI_OK_RESPONSE — Erfolg.

sonst — Fehlerfall.

Beispiel:

```
ord16                ret;
asi_device_info_t    asi_dev_info_var;
ret = asi_device_info(sizeof(asi_device_info_t),(ord8 *)&asi_dev_info_var);
```

Bemerkung:
Da sich dieser Aufruf nicht auf eine AS-I-Master-Anschaltung bezieht, sondern lediglich im AS-I-Treiber gehaltene Konfigurierungsdaten ausgelesen werden sollen, ist die Angabe eines Handles nicht notwendig. Der Aufruf kann schon vor dem ersten 'asi_init()'- und auch nach dem letzten 'asi_shut()'-Aufruf abgesetzt werden!

2.5.2.28 asi_cfg_in

Bedeutung

Mit diesem Aufruf werden alle Projektierdaten von der AS-I-Master-Anschaltung gelesen und in einer Datei abgelegt. Dazu werden die in den vorangegangenen Kapiteln beschriebenen Funktionen 'asi_get_permanent_parameter()', 'asi_get_permanent_configuration()', 'asi_get_lps()' und 'asi_get_flags()' benutzt.

Deklaration der Funktion:

```
ord16 asi_cfg_in(handle,filename)
void  *handle;
int8  *filename;
```

Beschreibung der Parameter:

handle	Handle als Rückgabewert des 'asi_init()'-Aufrufs.
filename	Name der Datei, in der die Projektierdaten abgelegt werden sollen. Diese Datei kann beim Aufruf 'asi_cfg_out()' wieder benutzt werden.

Rückgabewerte:

ASI_OK_RESPONSE	Erfolg.
sonst	Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret;

ret = asi_cfg_in(handle,(int8 *)"c:\sinec\cp2413.dw\data\asi_cfg1.dat");
```

2.5.2.29 asi_cfg_out

Bedeutung

Mit diesem Aufruf werden alle Projektierdaten aus einer Datei ausgelesen und auf die AS-I-Master-Anschaltung übertragen. Dazu werden die in den vorangegangenen Kapiteln beschriebenen Funktionen 'asi_set_permanent_parameter()', 'asi_set_permanent_configuration()', 'asi_set_lps()' und 'asi_set_operation_mode()' benutzt, wobei von den in der Datei gespeicherten Flags nur der Betriebsmodus gesetzt wird.

Deklaration der Funktion:

```
ord16 asi_cfg_out(handle,filename)
void  *handle;
int8  *filename;
```

Beschreibung der Parameter:

handle	Handle als Rückgabewert des 'asi_init()'-Aufrufs.
filename	Name der Datei, aus der die Projektierdaten ausgelesen werden sollen.

Rückgabewerte:

ASI_OK_RESPONSE	Erfolg.
sonst	Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret;

ret = asi_cfg_out(handle,(int8 *)"c:\sinec\cp2413.dw\data\asi_cfg1.dat");
```

2.5.2.30 asi_in_all

Bedeutung

Mit diesem Auftrag werden alle lesbaren AS-I-Daten und Listen in einem Aufruf ausgelesen. Innerhalb der AS-I-Library werden die folgenden Aufrufe benutzt, um die Daten von der Anschaltung abzufragen:

'asi_read_idi()', 'asi_get_permanent_parameter()', 'asi_read_parameter()', 'asi_get_permanent_configuration()', 'asi_read_actual_configuration()', 'asi_get_lps()', 'asi_get_las()', 'asi_get_lds()' und 'asi_get_flags()'.

Deklaration der Funktion:

```
ord16 asi_in_all(handle,buf_len,buffer)
void  *handle;
ord16 buf_len;
ord8  *buffer;
```

Beschreibung der Parameter:

handle — Handle als Rückgabewert des 'asi_init()'-Aufrufs.

buf_len — Dieser Parameter gibt die Länge des übergebenen Puffers an, der mindestens die Größe einer Variablen der nachfolgenden Struktur haben muß:

```
typedef struct asi_in_all_s
{           ord8  idi[ASI_MAX_NO_SLAVES];
      ord8  per_par[ASI_MAX_NO_SLAVES];
      ord8  actual_par[ASI_MAX_NO_SLAVES];
      ord8  per_conf[ASI_MAX_NO_SLAVES];
      ord8  actual_conf[ASI_MAX_NO_SLAVES];
      ord32 lps;
      ord32 las;
      ord32 lds;
ord16 flag;       }     asi_in_all_t;
```

Die Daten werden entsprechend den benutzten Aufrufen in der definierten Struktur abgelegt.

buffer — Zeiger auf den vom Anwender bereitgestellten Puffer. Die Daten werden entsprechend der obigen Struktur abgelegt.

Rückgabewerte:

ASI_OK_RESPONSE	Erfolg.
sonst	Fehlerfall.

Beispiel:

```
void          *handle;
ord16         ret;
asi_in_all_t  asi_in_all_var;
ret = asi_in_all(handle,sizeof(asi_in_all_t),(ord8 *)&asi_in_all_var);
```

2.5.2.31 asi_wd

Bedeutung

Mit diesem Auftrag wird eine Aktivitätsüberwachung (Watchdog) auf der AS-I-Master-Anschaltung aktiviert. Die Überwachungszeit kann vom Anwender gewählt werden. Defaultmäßig ist die Aktivitätskontrolle ausgeschaltet.

Ohne Aktivitätsüberwachung kann bei den Slaves ein gefährlicher Anlagenzustand entstehen, wenn

- ➢ das Anwenderprogramm nicht mehr korrekt arbeitet,
- ➢ der Rechner mit Hilfe der Tastenkombination '<CTRL><ALT><DEL>' zurückgesetzt wird.

Um diese Anlagenzustände zu vermeiden, sollten kritische Applikationen die Möglichkeit der Aktivitätsüberwachung nutzen und innerhalb der vorgegebenen Überwachungszeit Aufträge an den CP 2413 absetzen. Als Aufträge kommen alle Funktionen der AS-I-Library mit Ausnahme von 'asi_init()', 'asi_shut()' und 'asi_device_info()' in Frage. Läuft der Timer ab, so setzt die Baugruppe sämtliche Slaves zurück. Der Watchdog sollte dann deaktiviert werden, um weitere Aufträge absetzen zu können.

Der AS-I-Treiber achtet dabei darauf, daß je AS-I-Master-Anschaltung immer nur höchstens ein Prozeß die Aktivitätskontrolle einer AS-I-Master-Anschaltung nutzen kann. Ansonsten könnte ein Prozeß den Watchdog triggern, während ein anderer fehlerhaft arbeitet und gefährliche Anlagenzustände verursacht.

Deklaration der Funktion:

```
ord16 asi_wd(handle,timeout)
void  *handle;
ord16 timeout;
```

Beschreibung der Parameter:

handle — Handle als Rückgabewert des 'asi_init()'-Aufrufs.

timeout — Mit dem Wert 0 wird die Aktivitätskontrolle deaktiviert, mit den Werten 1-255 aktiviert. Andere Werte sind nicht erlaubt.

Wertebereich:

Wert	Bedeutung
0	Deaktivieren der Aktivitätskontrolle
1-255	Überwachungszeit in Sekunden
sonst	ungültig

Rückgabewerte:

ASI_OK_RESPONSE — Erfolg.

sonst — Fehlerfall.

Beispiel:

```
void    *handle;
ord16   ret;
ret = asi_wd(handle,5);
```

2.6 Systemspezifika unter MS-DOS und Windows

In diesem Abschnitt werden Sie mit Besonderheiten der AS-I-Schnittstelle unter MS-DOS und Windows vertraut gemacht. Sie erfahren, wie Sie eigene AS-I-Applikationen übersetzen und binden müssen.

2.6.1 Allgemeines

Die AS-I-Schnittstelle wird dem Anwender in Form von Libraries zur Verfügung gestellt. Sie werden für die **Memory-Modelle 'Large'** und **'Huge'** ausgeliefert und im folgenden Directory abgelegt:

➢ '<lw:>\sinec\cp2413.dw\lib' ('<lw:>' steht für das Laufwerk, auf dem Sie installiert haben).

Die für die Benutzung der Anwenderschnittstelle notwendigen Definitionen sind in folgender Datei hinterlegt:

➢ '<lw:>\sinec\cp2413.dw\inc\asi.h'

Es werden Libraries sowohl für MS-DOS als auch für Windows jeweils für verschiedene Compiler angeboten.

Die Namen dieser Libraries setzen sich wie folgt zusammen:

<Speichermodell><Betriebssystem>asi<Compiler>

mit

<Speichermodell>:	l	Large Model
	h	Huge Model
<Betriebssystem>:	d	MS-DOS
	w	Windows
<Compiler>:	msc	MSC-Compiler 7.0
	tc	Turbo-C-Compiler 1.0
	bc	Borland-C-Compiler 3.1

2.6.2 Ablage von Variablen

Normalerweise werden Variablen vom Compiler in einer Form im Speicher abgelegt, wie es für den Compiler am sinnvollsten erscheint. Hierbei können zwischen Komponenten einer Variablen Lücken entstehen (padding bytes).

☞ **Die Strukturen, die an der AS-I-Schnittstelle angeboten werden, sind so ausgelegt, daß mit Byte- oder Word-Alignment übersetzte Anwenderprogramme problemlos auf die einzelnen Komponenten zugreifen können. Double-Word-Alignment wird von der AS-I-Library nicht unterstüzt.**

In den folgenden Kapiteln werden Generierbeispiele aufgelistet, die Ihnen die notwendigen Compiler- und Linkeroptionen für Ihre Applikationen vermitteln sollen.

2.6.3 Laufwerks- und Pfadangaben

☞ **Die absoluten und auf Ihrem Zielrechner gültigen Laufwerks- und Pfadangaben müssen von Ihnen in die Generieranweisung eingebracht bzw. in den Suchpfad aufgenommen werden.**

2.6.4 Anwenderprogramme an- und abmelden

Anwenderprogramme müssen sich zwecks Kommunikation beim AS-I-Treiber anmelden, der zur Verwaltung Ressourcen belegt. Insgesamt sind 16 Anmeldungen gleichzeitig möglich. Wird eine Applikation durch die Tastenkombination '<CTRL>C' abgebrochen, bleiben die Ressourcen für den Prozeß reserviert, und die Anmeldung hat weiterhin Bestand. Um dies zu vermeiden, sollte im Anwenderprogramm ein '<CTRL>C'-Handler implementiert sein, der bei Programmabbruch sämtliche Abmeldungen beim AS-I-Treiber übernimmt.

☞ **Probleme mit Programmabbrüchen durch die Tastenkombination '<CTRL>C' sollten mit einem '<CTRL>C'-Handler umgangen werden.**

2.7 Übersetzen und Binden für MS-DOS

Arbeiten mit dem MSC-Compiler 7.0

Die AS-I-Library für den MSC-Compiler 7.0 hat unter MS-DOS den Namen 'ldasimsc.lib'. Das folgende Beispiel zeigt, wie ein Testprogramm 'test.c' mit dem Speichermodell 'Large' für MS-DOS übersetzt und gebunden wird:

```
cl /c /AL /Os /I\sinec\cp2413.dw\inc test.c

link @test.lnk
```

In der Datei 'test.lnk' stehen die Anweisungen für den Linker:

```
test.obj,
test.exe,
test.map,
\sinec\cp2413.dw\lib\ldasimsc.lib+
\msc70\lib\oldnames.lib+
\msc70\lib\llibce.lib
;
```

Arbeiten mit dem MS Visual C++ -Compiler 1.0

Wenn Sie den MS Visual C++ -Compiler 1.0 für Ihre MS-DOS-Applikationen einsetzen wollen, so können Sie dieselbe AS-I-Library wie für den MSC-Compiler 7.0 verwenden. Die Generieranweisungen für das Speichermodell 'Large' haben dann unter MS-DOS folgendes Aussehen:

```
cl /c /AL /Os /I\sinec\cp2413.dw\inc test.c
link @test.lnk
```

In der Datei 'test.lnk' stehen die Anweisungen für den Linker:

```
test.obj,
test.exe,
test.map,
\sinec\cp2413.dw\lib\ldasimsc.lib+
\msvc\lib\oldnames.lib+
\msvc\lib\llibce.lib
;
```

Arbeiten mit dem Turbo-C-Compiler 1.0

Für den Turbo-C-Compiler 1.0 liegen für MS-DOS zwei Ausprägungen der AS-I-Library vor: 'ldasitc.lib' für das Speichermodell 'Large' und 'hdasitc.lib' für das Speichermodell 'Huge'. Das folgende Beispiel zeigt, wie ein Testprogramm 'test.c' mit dem Speichermodell 'Large' für MS-DOS übersetzt und gebunden wird:

```
tcc -c -ml -I\sinec\cp2413.dw\inc test.c
tlink @test.lnk.
```

In der Datei 'test.lnk' stehen die Anweisungen für den Linker:

```
\tc10\lib\c0l.obj test.obj
test.exe
test.map
\tc10\lib\emu.lib \tc10\lib\mathl.lib \tc10\lib\cl.lib
\sinec\cp2413.dw\lib\ldasitc.lib
```

2.8 Übersetzen und Binden für Windows

Arbeiten mit dem MSC-Compiler 7.0

Die AS-I-Library für den MSC-Compiler 7.0 hat unter Windows den Namen 'lwasimsc.lib'. Das folgende Beispiel zeigt, wie ein Testprogramm 'test.c' mit dem Speichermodell 'Large' für Windows übersetzt und gebunden wird:

```
cl /c /AL /Os /I\sinec\cp2413.dw\inc test.c
link @test.lnk
```

In der Datei 'test.lnk' stehen die Anweisungen für den Linker:

```
test.obj,
test.exe,
test.map,
\sinec\cp2413.dw\lib\lwasimsc.lib+
\msc70\lib\oldnames.lib +
\msc70\lib\llibcew.lib +
\msc70\lib\libw.lib
;
```

Arbeiten mit dem MS Visual C++ -Compiler 1.0

Wenn Sie den MS Visual C++ -Compiler 1.0 für Ihre Windows-Applikationen einsetzen wollen, so können Sie dieselbe AS-I-Library wie für den MSC-Compiler 7.0 verwenden. Die Generieranweisungen für das Speichermodell 'Large' haben dann unter Windows folgendes Aussehen:

```
cl /c /AL /Os /I\sinec\cp2413.dw\inc test.c

link @test.lnk
```

In der Datei 'test.lnk' stehen die Anweisungen für den Linker:

```
test.obj,
test.exe,
test.map,
\sinec\cp2413.dw\lib\lwasimsc.lib+
\msvc\lib\oldnames.lib +
\msvc\lib\llibcew.lib +
\msvc\lib\libw.lib
;
```

Arbeiten mit dem Borland-C-Compiler 3.1

Die AS-I-Library für den Borland-C-Compiler 3.1 hat unter Windows den Namen 'lwasibc.lib'. Das folgende Beispiel zeigt, wie ein Testprogramm 'test.c' mit dem Speichermodell 'Large' für Windows übersetzt und gebunden wird:

```
bcc -c -ml -Os -I\sinec\cp2413.dw\inc test.c

tlink /Twe @test.lnk
```

In der Datei 'test.lnk' stehen die Anweisungen für den Linker:

```
\bc31\lib\c0wl.obj test.obj
test.exe
test.map
\sinec\cp2413.dw\lib\lwasibc.lib    \bc31\lib\mathwl.lib
\bc31\lib\import.lib \bc31\lib\cwl.lib
```

2.9 Besonderheiten für Windows

Die AS-I-Library steht für Windows-Applikationen auch in einer DLL-Version (dynamische Link-Library) zur Verfügung. Damit ist die AS-I-Funktionalität nutzbar für

- Programme der unterschiedlichsten Speichermodelle und
- 'Visual Basic'-Anwender.

Weiterhin wird die ASI-DLL nur einmal in den Hauptspeicher geladen, gleichgültig, wieviele Anwenderprogramme auf sie zugreifen.

Bei der Nutzung der DLL-Version der AS-I-Library ist auf folgendes zu achten:

- für die AS-I-Funktionen müssen die 'PASCAL'-Aufrufkonvention gültig sein,
- es muß sowohl mit 'far'-Calls als auch mit 'far'-Data gearbeitet werden.

  Bei den Prototypings der AS-I-Funktionen in der Headerdatei '\sinec\cp2413.dw\inc\asi.h' wurde diese Anpassung mit Hilfe der Defines 'ASI_FAR_DATA' und 'ASI_FAR_PASCAL' bereits berücksichtigt und kann mit Hilfe des Defines 'ASI_DLL' aktiviert werden.

☞ **Bei der Verwendung der ASI-DLL in 'C'-Programmen ist darauf zu achten, daß alle Module, die AS-I-Funktionen aufrufen, mit dem Define 'ASI_DLL' übersetzt werden, und daß die entsprechenden Importlibraries zur Anwendung gebunden werden. Bei allen Aufrufen der ASI-DLL handelt es sich um 'far'-Calls, bei Pointern um 'far'-Data-Pointer (wichtig für das Handle vom Typ 'void ASI_FAR_DATA *'). Die ASI-DLL muß ins aktuelle Arbeitsverzeichnis oder ins Windows-System-Directory kopiert werden.**

Arbeiten mit Visual Basic 2.0

Die AS-I-Aufrufe haben für 'Visual Basic'-Anwender ein etwas modifiziertes Aussehen:

- der Typ des Handles ist 'LONG' (anstelle 'void far *');
- Übergabeparameter vom Typ 'Pointer' werden als 'As Any' aufgefaßt.

Diese Anpassungen sind notwendig, da diese Grundtypen aus der Programmiersprache 'C' von 'Visual Basic' nicht unterstützt werden.

Die AS-I-Funktionen werden in der Datei '\sinec\cp2413.dw\inc\asidec.bas' auf Ihrem Zielrechner deklariert. Diese Deklarationen können auch als Prototypings benutzt werden. Die für 'Visual Basic'-Anwendungen notwendigen Definitionen sind in den Datein '\sinec\cp2413.dw\inc\asiconst.bas' und '\sinec\cp2413.dw\inc\asitypes.bas' enthalten.

☞ **Die Dateien 'asidec.bas', 'asiconst.bas' und 'asitypes.bas' sollten Bestandteil Ihres Projects unter 'Visual Basic' sein.**

Arbeiten mit dem MSC-Compiler 7.0

Die Import-Library der ASI-DLL für den MSC-Compiler 7.0 hat unter Windows den Namen 'asimsc.lib'. Das folgende Beispiel zeigt, wie ein Testprogramm 'test.c' mit dem Speichermodell 'Small' für Windows übersetzt und gebunden wird:

```
cl /c /AS /Os /I\sinec\cp2413.dw\inc test.c -DASI_DLL
link /NOD @test.lnk
```

In der Datei 'test.lnk' stehen die Anweisungen für den Linker:

```
test.obj,
test.exe,
test.map,
\sinec\cp2413.dw\lib\asimsc.lib+
\msc70\lib\oldnames.lib +
\msc70\lib\slibcew.lib +
\msc70\lib\libw.lib
test.def;
```

Die Modul-Definitions-Datei 'test.def' hat folgenden Aufbau:

```
NAME        TEST
EXETYPE     WINDOWS
CODE        PRELOAD MOVEABLE DISCARDABLE
DATA        PRELOAD MOVEABLE MULTIPLE
HEAPSIZE    1024
EXPORTS

;===List your explicity exported functions here ===
```

Arbeiten mit dem MS Visual C++ -Compiler 1.0

Wenn Sie den MS Visual C++ -Compiler 1.0 für Ihre MS-DOS-Applikationen einsetzen wollen, so können Sie dieselbe Import-Library wie für den MSC-Compiler 7.0 verwenden. Das folgende Beispiel zeigt, wie ein Testprogramm 'test.c' mit dem Speichermodell 'Small' für Windows übersetzt und gebunden wird:

```
cl /c /AS /Os /I\sinec\cp2413.dw\inc test.c -DASI_DLL
link /NOD @test.lnk
```

In der Datei 'test.lnk' stehen die Anweisungen für den Linker:

```
test.obj,
test.exe,
test.map,
\msvc\lib\oldnames.lib +
\msvc\lib\libw.lib+
\msvc\lib\slibcew.lib +
\sinec\cp2413.dw\lib\asimsc.lib
test.def;
```

Die Modul-Definitions-Datei 'test.def' hat folgenden Aufbau:

```
NAME        TEST
EXETYPE     WINDOWS
CODE        PRELOAD MOVEABLE DISCARDABLE
DATA        PRELOAD MOVEABLE MULTIPLE
HEAPSIZE    1024
EXPORTS
;===List your explicity exported functions here ===
```

Arbeiten mit dem Borland-C-Compiler 3.1

Die Import-Library der ASI-DLL für den Borland-C-Compiler 3.1 hat unter Windows den Namen 'asibc.lib'. Das folgende Beispiel zeigt, wie ein Testprogramm 'test.c' mit dem Speichermodell 'Large' für Windows übersetzt und gebunden wird:

```
bcc -c -ml -Os -I\sinec\cp2413.dw\inc -DASI_DLL

tlink /Twe @test.lnk
```

In der Datei 'test.lnk' stehen die Anweisungen für den Linker:

```
\bc31\lib\c0wl.obj test.obj
test.exe
test.map
\sinec\cp2413.dw\lib\asibc.lib \bc31\lib\mathwl.lib
\bc31\lib\import.lib \bc31\lib\cwl.lib
```

2.10 Hinweise und Fehlermeldungen

Sie kennen nun "AS-I-2413 MS-DOS/Windows". Sie haben gelernt, die Hard- und Software zu installieren und zu konfigurieren sowie die AS-I-Library zu Ihren Applikationen zu binden. Bei all diesen Tätigkeiten können jedoch auch kleine Probleme auftreten. Dieses Kapitel soll Ihnen helfen, Schwierigkeiten zu bewältigen, welche bei der Handhabung von "AS-I-2413 MS-DOS/Windows" auftreten können.

Die Komponenten zu "AS-I-2413 MS-DOS/Windows" liefern Fehlermeldungen, die auf die Ursache des Problems hinweisen. Im folgenden wird beschrieben, was Sie tun können, um das Problem zu lösen.

2.10.1 Das DPRAM

Der CP 2413 benötigt ein DPRAM der Größe 2K Byte, das aus dem Speicherbereich von C8000h bis EFFFFh stammen muß, unter den in Kapitel 2.4.3 genannten Bedingungen. Bei manchen Rechnern liegt das Shadow RAM oder der Cache auf dem Bereich E0000h. Diesen Bereich können Sie für den CP 2413 nutzen, wenn Sie im Systemsetup Cache und Shadow RAM ausschalten (Cache disabled, Shadow RAM disabled).

Der vom CP 2413 benutzte Speicherbereich darf nicht von Expanded Memory Managern, wie z.B. EMM386, QUEM oder 386MAX verwendet werden. Beim Starten dieser Manager müssen Sie dann angeben, daß dieser Bereich nicht benutzt werden darf.

Beispiel für EMM386:

```
device=emm386.sys     .....       X=C800-C87F
```

Dies bedeutet, daß EMM386 den Bereich von C8000h bis C87FFh nicht benutzen soll.

2.10.2 Status- und Fehlermeldungen des AS-I-Treibers

Der AS-I-Treiber gibt beim Installieren die nachfolgenden Fehler- und Statusmeldungen auf dem Bildschirm aus:

ASI driver already loaded.

Der AS-I-Treiber ist bereits geladen und unterstützt bis zu vier AS-I-Master-Anschaltungen. Ein mehrmaliges Laden des Treibers ist nicht möglich. Bitte überprüfen Sie, ob Ihre Anforderungen nicht vom bereits installierten AS-I-Treiber erfüllt werden können.

ASI driver Vx.yy dd.mm.yy installed. (c) Siemens AG.

Der AS-I-Treiber mit der Versionsnummer x.yy und dem Datum dd.mm.yy konnte erfolgreich installiert werden. Bitte stellen Sie mittels AS-I-Monitor die Betriebsbereitschaft Ihrer AS-I-Master-Anschaltung(en) her.

Given interrupt number not valid or syntax error.

Der beim Aufruf des Treibers übergebene Parameter konnte nicht als Interruptnummer interpretiert werden, liegt nicht im Bereich von 80h bis c0h oder ist nicht geradzahlig. Bitte überprüfen Sie den Übergabewert auf Korrektheit und Zulässigkeit.

Interrupt can't be used.

Die beim Aufruf des Treibers übergebene Interruptnummer ist schon belegt. Bitte überprüfen Sie, ob nicht ein anderer Interrupt zur Verfügung steht.

There is no free interrupt vector.

Alle Interrupts von 80h bis c0h sind belegt. Bitte entfernen Sie andere Programme oder Treiber, die einen dieser Interrupts belegen.

2.10.3 Status- und Fehlermeldungen des AS-I-Monitors

Der AS-I-Monitor gibt nach dem Starten folgende Status- und Fehlermeldungen auf dem Bildschirm aus:

ASI driver not found.
Der AS-I-Monitor konnte den AS-I-Treiber nicht finden. Bitte starten Sie den AS-I-Treiber unter Berücksichtigung der Fehler- und Statusmeldungen nach Kapitel 2.10.2.

ASI driver not removed, error code = x.
Die AS-I-Treiber konnte nicht beendet werden, bzw. der vom AS-I-Treiber belegte Speicher konnte nicht freigegeben werden. 'x' gibt dabei den Fehlercode an, der mit einem der Returnwerte der AS-I-Library nach Kapitel 2.10.4 identisch ist.

ASI driver removed.
Die konfigurierten AS-I-Master-Anschaltungen wurden in den Offline-Betrieb geschaltet, und der AS-I-Treiber wurde aus dem Hauptspeicher entfernt. Damit ist bis zum erneuten Laden des AS-I-Treibers keinerlei Kommunikation über die AS-I-Baugruppen mehr möglich.

DPRAM already used.
Das zu verwendende DPRAM wird bereits bei einer anderen AS-I-Master-Anschaltung benutzt. Bitte wählen Sie einen anderen Speicherbereich aus, unter Berücksichtigung der DPRAMs sowohl anderer AS-I-Master-Anschaltungen als auch anderer SINEC- oder Fremdbaugruppen.

Error converting preloader in file x.
Bei der Aufbereitung des Preloaders in Datei 'x' ist ein Fehler aufgetreten. Bitte installieren Sie nochmals von der Installationsdiskette.

Error reading file x.
Beim Lesen aus der Datei 'x' ist ein Fehler aufgetreten. Vergewissern Sie sich, daß auf die Datei lesend zugegriffen werden kann, oder installieren Sie nochmals von der Installationsdiskette.

File x not found.
Die Datei names 'x' konnte nicht gefunden werden. Der AS-I-Monitor greift auf die Preloader-Datei und die Firmware-Datei zu, die beim Installieren auf Ihren Zielrechner kopiert wurden. Bitte überprüfen Sie die Laufwerksangabe oder installieren Sie nochmals von der Installationsdiskette.

Firmware loaded.
Die Firmware wurde auf die AS-I-Master-Anschaltung geladen. Die Baugruppe ist jetzt betriebsbereit, und Anwenderprogramme können über die AS-I-Library Aufträge an die Anschaltung absetzen.

Firmware not loaded, error code = x.
Die Firmware konnte nicht geladen werden. 'x' gibt dabei den Fehlercode an, der mit einem der Returnwerte der AS-I-Library nach Kapitel 2.10.4 identisch ist.

I/O port already used.
Der zu verwendende I/O-Adressraum wird bereits bei einer anderen AS-I-Master-Anschaltung benutzt. Bitte wählen Sie ein anderes I/O-Port aus, unter Berücksichtigung der I/O-Ports sowohl anderer AS-I-Master-Anschaltungen als auch anderer SINEC- oder Fremdbaugruppen.

Status- und Fehlermeldungen des AS-I-Monitors (Fortsetzung)

Max. number of boards configured.

Die maximale Anzahl von AS-I-Master-Anschaltungen wurde bereits konfiguriert. Der AS-I-Treiber unterstützt bis zu vier Anschaltungen.

Name too long.

Der Ansprechname für die AS-I-Master-Anschaltung ist zu lang. Bitte gewährleisten Sie, daß der Name nicht länger ist als 8 Zeichen.

Not enough memory.

Der AS-I-Monitor konnte nicht genügend Speicher allokieren. Bitte entfernen Sie nicht benötigte Treiber oder speicherresidente Programme aus dem Hauptspeicher.

No ok. from board.

Die Firmware wurde nicht korrekt auf die Anschaltung geladen bzw. konnte sich nicht korrekt beim AS-I-Monitor melden. Bitte überprüfen Sie nochmals die Hardwareabhängigkeiten wie DPRAM und Konfigurationsregister. Unter Umständen ist eine Neuinstallation notwendig.

Preloader in file x too long.

Die Preloaderdatei 'x' ist zu groß. Bitte installieren Sie nochmals von der Installationsdiskette.

Value of DPRAM unknown.

Der als Wert für das Anfangssegment des DPRAMs angegebene Parameter ist nicht zulässig. Bitte beachten Sie, daß nur die Werte von c800 bis ef80 in Schritten von 80 möglich sind.

Value of I/O port unknown.

Der als Wert für das Basis-Konfigurationsregister angegebene Parameter ist nicht zulässig. Bitte überprüfen Sie, welches der Basis-Konfigurationsregister 0x0200, 0x0208, 0x0280, 0x0288, 0x0290, 0x0320, 0x03E0 oder 0x03E8 Sie verwenden können.

Neben diesen Fehler- und Statusmeldungen gibt der AS-I-Monitor noch eine Usage-Meldung auf dem Bildschirm aus, falls die Aufrufsyntax nicht eingehalten wurde. In diesem Fall empfiehlt es sich, die Anmerkungen zum Thema Softwareinstallation in Kapitel 2.4 dieses Handbuchs nachzulesen.

2.10.4 Returnwerte der AS-I-Aufrufe

Im folgenden sind die Returnwerte der AS-I-Aufrufe und ihre Bedeutung anhand der dafür vorgesehenen Defines aufgeführt. Die Konstanten sind in der Datei 'asi.h' definiert, die sich im Directory '<lw:>\sinec\cp2413.dw\inc' auf Ihrem Zielrechner befindet. '<lw:>' steht dabei für das Laufwerk, auf dem Sie die Software installiert haben.

ASI_COMMAND_NOT_ALLOWED

Der von Ihnen abgesetzte AS-I-Master-Aufruf ist im augenblicklichen Zustand der Firmware nicht erlaubt. Bitte schalten Sie auf den laut /1/ für Ihren Auftrag notwendigen Betriebs- oder Offline-Modus um.

ASI_DATA_NOT_VALID

Die vom angesprochenen Slave gelieferten Daten sind nicht gültig. Dieser Fehler kann nur beim Aufruf 'asi_execute_command()' auftreten. Bitte überprüfen Sie die Existenz und die Funktionsfähigkeit des Slaves.

Die Funktion liefert als Datum eine der nachfolgenden Fehlerursachen zurück (siehe auch /1/):

ASI_I0_BIT_ERROR

Die Übertragung des I0-Bits war nicht störungsfrei.

ASI_I1_BIT_ERROR

Die Übertragung des I1-Bits war nicht störungsfrei.

ASI_I2_BIT_ERROR

Die Übertragung des I2-Bits war nicht störungsfrei.

ASI_I3_BIT_ERROR

Die Übertragung des I3-Bits war nicht störungsfrei.

ASI_PARITY_BIT_ERROR

Die Übertragung des Parity-Bits war nicht störungsfrei.

ASI_PARITY_ERROR

Bei der Übertragung wurde ein Parity-Fehler festgestellt.

ASI_SLAVE_TIMEOUT

Der angesprochene Slave antwortete nicht innerhalb von 3 ASI-Zyklen.

ASI_START_BIT_ERROR

Die Übertragung des Start-Bits war nicht störungsfrei.

ASI_STOP_BIT_ERROR

Die Übertragung des Stop-Bits war nicht störungsfrei.

ASI_TRANSMISSION_GAP_ERROR

Die Masterpause war nicht störungsfrei.

ASI_DRIVER_NOT_FOUND

Der AS-I-Treiber konnte nicht gefunden werden. Bitte bringen Sie Ihre Anschaltung(en) mit Hilfe des Batches '<lw:>\sinec\cp2413.dw\bin\startasi.bat' in Betriebsbereitschaft unter Berücksichtigung der Fehler- und Statusmelden des AS-I-Treibers und des AS-I-Monitors.

Returnwerte der AS-I-Aufrufe (Fortsetzung)

ASI_DRV_INIT_MAX

Die maximale Anzahl von Anmeldungen beim Kommunikationssystem (der AS-I-Treiber unterstützt bis zu 16 Anmeldungen gleichzeitig) wurde bereits erfolgreich durchgeführt und hat noch Bestand. Bitte überprüfen Sie, ob Sie nicht ein bereits für Ihre Applikation vergebenes Handle benutzen können, oder ob sich nicht ein anderer Prozeß beim Kommunikationssystem abmelden kann.

ASI_ERROR_DELETING_ADDRESS

Es ist ein Fehler beim Löschen der Betriebsadresse des von Ihnen angesprochenen Slaves aufgetreten. Bitte überprüfen Sie die Funktionsfähigkeit des Slaves.

ASI_ERROR_FILE_READ

Die angegebene Datei konnte nicht ausgelesen werden. Bitte überprüfen Sie die Zugriffsrechte und -konflikte durch andere Prozesse.

ASI_ERROR_FILE_WRITE

Die angegebene Datei konnte nicht geschrieben werden. Bitte überprüfen Sie die Dateiattribute sowie Zugriffsrechte und -konflikte durch andere Prozesse.

ASI_ERROR_SETTING_ADDRESS

Es ist ein Fehler beim Setzen der Betriebsadresse des von Ihnen angesprochenen Slaves aufgetreten. Bitte überprüfen Sie die Funktionsfähigkeit des Slaves.

ASI_ERROR_STORING_ADDRESS

Der von Ihnen angesprochene Slave konnte die neue Betriebsadresse nicht nicht-flüchtig speichern. Bitte überprüfen Sie die Funktionsfähigkeit des Slaves.

ASI_FILE_INCONSISTENT

Die angegebene Datei ist nicht konsistent. Dieser Fehler tritt nur beim AS-I-Aufruf 'asi_cfg_out()' auf. Stellen Sie sicher, daß die angegebene Datei mit dem Aufruf 'asi_cfg_in()' erstellt und danach nicht manipuliert wurde.

ASI_FILE_NOT_FOUND

Die beim AS-I-Aufruf angegebene Datei wurde nicht gefunden. Bitte überprüfen Sie die Laufwerksangabe, den Dateinamen und die Dateiattribute (eventuell könnte es sich auch um eine versteckte Datei handeln).

ASI_LIST_EMPTY

Die zum Projektieren der Slaves übergebene Liste (LPS) ist leer. Bitte wählen Sie mindestens einen Slave aus, den Sie projektieren möchten.

ASI_MEM

Die AS-I-Library konnte nicht genügend temporären Speicher allokieren. Bitte entfernen Sie nicht benötigte Treiber, speicherresidente Programme oder Applikationen aus dem Hauptspeicher.

ASI_NOCONFIG

Die ausgewählte Anschaltung ist nicht konfiguriert. Bitte bringen Sie die Anschaltung durch den AS-I-Monitor in einen betriebsbereiten Zustand.

ASI_OK_RESPONSE

Der Auftrag wurde erfolgreich abgeschlossen.

Returnwerte der AS-I-Aufrufe (Fortsetzung)

ASI_PARAM_ADDRESS

Die im Aufruf übergebene Adresse ist nicht gültig. Es dürfen nur Werte von 0 bis 31 (jeweils einschließlich) und der Wert für den Listenmodus (ASI_LIST) übergeben werden.

ASI_PARAM_CMD

Der im 'asi_execute_command()'-Aufruf als Kommando übergebene Werte ist unzulässig. Bitte verwenden Sie nur die Definitionen aus der Header-Datei 'asi.h'.

ASI_PARAM_DATA

Die vorgegebenen Daten sind ungültig. Bitte überprüfen Sie die Daten entsprechend des ausgewählten AS-I-Aufrufs auf Zulässigkeit. Bei Projektierdaten z.B. müssen die Bits 4 bis 7 mit 0 vorbelegt sein.

ASI_PARAM_HANDLE

Das im AS-I-Aufruf übergebene Handle konnte nicht als AS-I-Handle identifiziert werden. Bitte vergleichen Sie diesen Parameter mit dem Rückgabewert der Anmeldefunktion 'asi_init()'.

ASI_PARAM_MODE

Der für die Funktionen 'asi_set_operation_mode()' bzw. 'asi_set_offline_mode()' angegebene Wert für den Modus ist nicht zulässig. Es kann lediglich zwischen dem geschützten Betriebsmodus und dem Projektierungsmodus, bzw. dem Online- und dem Offline-Betriebsmodus umgeschaltet werden. Für die einzelnen Modi stehen in der Header-Datei 'asi.h' entsprechende Definitionen zur Verfügung.

ASI_PARAM_NAME

Der im Aufruf 'asi_init()' angegebene Name für den ausgewählten CP ist unbekannt. Er darf maximal 8 Zeichen lang sein und muß mit einem Namen übereinstimmen, der bei der Inbetriebsnahme in der Datei 'startasi.bat' eingetragen wurde.

ASI_PARAM_TIMEOUT

Die gewählte Überwachungszeit für die Aktivitätskontrolle ist unzulässig. Es darf nur ein Wert aus dem Bereich von 0 bis einschließlich 255 verwendet werden.

ASI_SLAVE0_ALREADY_EXISTS

Sie haben versucht, die Adresse eines Slaves zu verändern, obwohl Slave 0 existiert. Weisen Sie diesem Slave eine gültige Adresse zu (Hinweis: Eine Adressänderung erfolgt immer über Slave 0 (/1/)).

ASI_SLAVE0_EXISTS

Es ist nicht möglich, in den geschützten Betrieb umzuschalten, wenn Slave 0 existiert. Bitte weisen Sie diesem Slave eine gültige Adresse zu.

ASI_SLAVE_ADDRESS1_NOT_FOUND

Sie haben versucht, die Adresse eines nicht existierenden Slaves zu verändern. Bitte überprüfen Sie die Adress-Angaben im AS-I-Aufruf.

ASI_SLAVE_ADDRESS1_NOT_VALID

Sie haben versucht, Parameter oder Konfigurationsdaten für Slave 0 zu projektieren oder Parameter an Slave 0 zu übertragen. Dies ist unzulässig. Bitte überprüfen Sie die ausgewählte Slave-Adresse.

Returnwerte der AS-I-Aufrufe (Fortsetzung)

ASI_SLAVE_ADDRESS2_ALREADY_EXISTS

Sie haben versucht, einem Slave eine bereits existierende Adresse zuzuweisen. Bitte überprüfen Sie die Adress-Angaben im AS-I-Aufruf.

ASI_SLAVE_ADDRESS2_NOT_VALID

Die im Aufruf übergebene Adresse ist nicht gültig. Es dürfen nur Werte von 0 bis 31 (jeweils einschließlich) übergeben werden.

ASI_SLAVE_NOT_IN_LAS

Der Auftrag wurde abgelehnt, weil der ausgewählte Slave zum Zeitpunkt der Auftragsbearbeitung nicht in der LAS aufgenommen war. Bitte überprüfen Sie die Existenz des von Ihnen angesprochenen Slaves und seine Funktionsfähigkeit.

ASI_TIMEOUT

Die AS-I-Master-Anschaltung meldet sich nicht innerhalb einer Sekunde. In diesem Fall sollten Sie die Kommunikation beenden, den AS-I-Treiber aus dem Speicher entfernen, neu laden und die Anschaltung(en) in Betriebsbereitschaft bringen.

ASI_TRANSMISSION_ERROR

Bei einer AS-I-Nachricht ist ein Übertragungsfehler aufgetreten. Bitte überprüfen Sie die Existenz des von Ihnen angesprochenen Slaves und seine Funktionstüchtigkeit.

ASI_USERMEM

Der beim AS-I-Aufruf übergebene Speicher ist nicht ausreichend groß. Bitte stellen Sie genügend Puffer für den Auftrag zur Verfügung und wiederholen Sie den Aufruf nochmals. Informationen über den Pufferbedarf ist beim jeweiligen AS-I-Aufruf in Kapitel 2.5 nachzulesen.

ASI_WATCHDOG_TIMER_RESET

Der Watchdog der selektierten Baugruppe wurde zwar aktiviert, allerdings wurden während der eingestellten Überwachungszeit keine Aufträge an die Firmware übergeben. Die Firmware hat in den Offline-Betrieb umgeschaltet. Bitte stellen Sie den ursprünglichen Zustand der Firmware durch die entsprechenden AS-I-Aufrufe wieder her.

ASI_WD_IN_USE

Der Watchdog der ausgewählten Anschaltung wurde bereits vom gleichen Prozeß oder von einer anderen Applikation aktiviert. Bitte beachten Sie, daß der Watchdog einer Anschaltung nicht mehrfach aktiviert werden darf.

ASI_WD_NOT_USED

Der Watchdog der ausgewählten Anschaltung wurde nicht aktiviert bzw. wurde bereits erfolgreich deaktiviert. Eine Aktivitätsüberwachung für die Anschaltung besteht nicht (mehr).

❑

A ASI Protocol Implementation Conformance Statement (PICS)

A.1 PICS für CP 2413

Hersteller	Siemens AG
Produkt Name	CP 2413 - SINEC S1 Masterbaugruppe für AT-PCs
Bestellnummer	6GK1241-3SA00
Ausgabestand	1
Master Profil	M1
Datum	27.9.94

Liste der verfügbaren Master-Funktionen

Nr.	Funktion oder Aufruf an der Host-Schnittstelle (Symbolische Darstellung)	M1	Bemerkung / Realisierung der Funktion durch / Kapitel
1	Image, Status = Read_IDI()	X	asi_read_idi() / 2.5.2.3
2	Status = Write_ODI(Image)	X	asi_write_odi() / 2.5.2.4
3	Status = Set_Permanent_Parameter(Addr, Param)	X	asi_set_permanent_parameter / 2.5.2.5
4	Param, Status = Get_Permanent_Parameter(Addr)	X	asi_get_permanent_parameter / 2.5.2.6
5	Status, GParam = Write_Parameter(Addr, Param)	X	asi_write_parameter() / 2.5.2.7
6	Status, Param = Read_Parameter(Addr)	X	asi_read_parameter() / 2.5.2.8
7	Status = Store_Actual_Parameters()	X	asi_store_actual_parameters() / 2.5.2.9
8	Status = Set_Permanent_Configuration(Addr, Config)	X	asi_set_permanent_configuration / 2.5.2.10
9	Status, Config = Get_Permanent_Configuration(Addr)	X	asi_get_permanent_configuration / 2.5.2.11
10	Status = Store_Actual_Configuration()	X	asi_store_actual_configuration / 2.5.2.12. Bei diesem Kommando wird zusätzlich ein Neustart des CPs durchgeführt.
11	Status, Config = Read_Actual_Configuration(Addr)	X	asi_read_actual_configuration / 2.5.2.13
12	Status = Set_LPS(List31)	X	asi_set_lps() / 2.5.2.14
13	Status, List31 = Get_LPS()	X	asi_get_lps() / 2.5.2.15
14	Status, List31 = Get_LAS()	X	asi_get_las() / 2.5.2.16

15	Status, List32 = Get_LDS()	X	asi_get_lds() / 2.5.2.17
16.0	Status = Get_Flags()	X	asi_get_flags() / 2.5.2.18
16.1	Status, Flag = Get_Flag_Config_OK()	X	asi_get_flags() / 2.5.2.18
16.2	Status, Flag = Get_Flag_LDS.0()	X	asi_get_flags() / 2.5.2.18
16.3	Status, Flag = Get_Flag_Auto_Address_Assign()	X	asi_get_flags() / 2.5.2.18
16.4	Status, Flag = Get_Flag_Auto_Prog_Available()	X	asi_get_flags() / 2.5.2.18
16.5	Status, Flag = Get_Flag_Configuration_Active()	X	asi_get_flags() / 2.5.2.18
16.6	Status, Flag = Get_Flag_Normal_Operation_Active()	X	asi_get_flags() / 2.5.2.18
16.7	Status, Flag = Get_Flag_APF()	X	asi_get_flags() / 2.5.2.18
16.8	Status, Flag = Get_Flag_Offline_Ready()	X	asi_get_flags() / 2.5.2.18
17	Status = Set_Operation_Mode(Mode)	X	asi_set_operation_mode() / 2.5.2.19
18	Status = Set_Offline_Mode(Mode)	X	asi_set_offline_mode() / 2.5.2.20
19	Status = Activate_Data_Exchange(Mode)	-	nicht implementiert
20	Status = Change_Slave_Address(Addr1, Addr2)	X	asi_change_slave_address() / 2.5.2.21
21.1	Status, Resp = Cmd_Reset_ASI_Slave(Addr, RESET)	X	asi_execute_command() / 2.5.2.23
21.2	Status, Resp = Cmd_Read_IO_Configuration(Addr, CONF)	X	asi_execute_command() / 2.5.2.23
21.3	Status, Resp = Cmd_Read_Identification_Code(Addr, IDCOD)	X	asi_execute_command() / 2.5.2.23
21.4	Status, Resp = Cmd_Read_Status(Addr, STAT)	X	asi_execute_command() / 2.5.2.23
21.5	Status, Resp = Cmd_Read_Reset_Status(Addr, STATRES)	X	asi_execute_command() / 2.5.2.23

Zeichenerklärung zu Spalte 3

Zeichen	Bedeutung
X	Funktion vorhanden
-	Funktion nicht vorhanden

Abhängigkeit der ASI Zykluszeit von der Anzahl der angeschlossenen Slaves

Die maximale Zykluszeit beträgt <= 5 ms bei 31 Slaves. Diese maximale Zykluszeit wird auch bei geringerer Anzahl von Slaves nicht unterschritten.
❒

B Literaturverzeichnis

/1/ ASI Das Aktuator-Sensor-Interface für die Automation

Werner Kriesel, O.W. Madelung, Carl Hanser Verlag München Wien 1994

/2/ ASI Complete Specification

beziehbar über ASI-Verein e.V.

die Anschrift lautet: ASI-Verein e.V.
Auf den Broich 4A
51519 Odenthal

Tel.: 02174 / 40756
Fax.:02174 / 41571

(Die AS-I-Technologie wird durch den ASI-Verein e.V. gefördert)

/3/ SINEC Industrielle Kommunikationsnetze

Katalog IK 10

/4/ Hantierungsbausteine sind jeweils beschrieben in:

Für S5-115 U als Bestandteil des Gerätehandbuches
Bestnr.: 6 ES 5998-3UF x1*) für CPU 945
Bestnr.: 6 ES 5998-0UF x3*) für CPU 941 - CPU 944

Für S5-135 U beziehbar als Paket HTB-Software + Beschreibung
Bestnr.: 6 ES 5842-7CB 01 für CPU 928 A/B - CPU 948

Für S5-155 U beziehbar als Paket HTB-Software + Beschreibung
Bestnr.: 6 ES 5846-7CA 01 für CPU 946 / 947

*) x: beliebig

C Abkürzungen / Begriffe

- **AG** Automatisierungsgerät; z.B. SPS SIMATIC S5
- **AS-I** Aktor-Sensor-Interface
- **APF** ASI-Power-Fail. Flag bzw. LED-Anzeige, die kennzeichnet, daß die Versorgungsspannung auf der AS-I-Leitung zu niedrig oder ausgefallen ist. (z.B. Ausfall des AS-I-Netzteils)
- **AS-I-Library** Library, mittels derer Anwendungsprogramme mit dem ASI-Treiber kommunizieren können.
- **AS-I-Treiber** Treiber, der Anwenderprogrammen die Leistungen des CP 2413 zugänglich macht.
- **CP** Communication Processor: Kommunikationsbaugruppe für den Einbau in Rechner oder Automatisierungsgeräte.
- **CP 2433** Kommunikationsanschaltung für SIMATIC S5 und ET 200 an SINEC S1.
- **CP 2430** Kommunikationsanschaltung für SIMATIC S5 S5-115U, S5-135U und S5-155U an SINEC S1.
- **CP 1413** Kommunikationsanschaltung für PCs und PGs an SINEC H1
- **CP 2413** Kommunikationsanschaltung für PCs und PGs an SINEC S1
- **CP 5412** Kommunikationsanschaltung für PCs und PGs an SINEC L2
- **FW** Firmware, hier auf dem CP 2413 ablaufende Software
- **LAS** Liste der aktivierten Slaves
- **LDS** Liste der erkannten Slaves
- **LPS** Liste der projektierten Slaves
- **PG** Programmiergerät
- **SINEC** 'Siemens Network Architecture for Automation and Engineering', Produktlinie für industrielle Kommunikation von Siemens.❑

D Hinweise zur CE-Kennzeichnung

Produkt-bezeichnung: CP 2413 6GK1 241-3SA00

EU-Richtlinie EMV 89/336/EWG

Das obige SINEC-Produkt erfüllt die Anforderungen der EU-Richtlinie 89/336/EWG "Elektromagnetische Vertäglichkeit".

CE

Die EU-Konformitätserklärung wird gemäß der obengenannten EU-Richtlinie für die zuständigen Behörden zur Verfügung gehalten bei:

Siemens Aktiengesellschaft
Bereich Automatisierungstechnik
Industrielle Kommunikation (AUT 93)
Postfach 4848
D-90327 Nürnberg

Einsatzbereich

Das Produkt erfüllt folgende Anforderungen:

Einsatzbereich	Anforderungen an	
	Störaussendung	Störfestigkeit
Industrie	EN 50081-2 : 1993	EN 50082-2 : 1995

Das Produkt ist mit einer Einzelgenehmigung auch einsetzbar im Wohnbereich (Wohnbereich, Geschäfts- und Gewerbebereiche sowie Kleinbetriebe).

Einsatzbereich	Anforderungen an	
	Störaussendung	Störfestigkeit
Wohnbereich	Einzelgenehmigung	EN 50082-1 : 1992

Die Einzelgenehmigung müssen Sie bei einer Behörde oder Prüfstelle einholen.
In Deutschland erteilt die Einzelgenehmigung das Bundesamt für Post und Telekommunikation und seine Nebenstellen.

Aufbaurichtlinien beachten

Das Produkt erfüllt die Anforderungen, wenn Sie bei Installation und Betrieb die Aufbaurichtlinien einhalten, die in folgenden Dokumentationen beschrieben sind:

1. SINEC S1 CPs Gerätehandbuch;
2. Einbauanweisung im Handbuch Ihres Endgerätes.

Arbeiten am Produkt

Zum Schutz des Produktes vor Entladung von statischer Elektrizität muß sich das Bedienpersonal vor dem Berühren der Baugruppe elektrostatisch entladen.

Hinweis

Das Produkt wurde in einem Gerät getestet, das ebenfalls die oben genannten Normen einhält. Beim Betrieb der Baugruppe in einem Gerät, das diese Normen nicht erfüllt, kann die Einhaltung der entsprechenden Werte nicht garantiert werden.

SIEMENS AG
AUT 933
Siemensallee 84

76187 Karlsruhe
Federal Republic of Germany

Vorschläge Korrekturen
Suggetions Corrections

Für Druckschrift bzw. Handbuch
For instruction or manual

Titel/title

SINEC S1 Masterbaugruppe 2413

Absender/From - Name/Name

Firma/Dienststelle - Company/Department

Anschrift/Address

Bestell-Nr./Order No.

6GK 1971-2SA01-0AA0

Telefon/Telephone

Vorschläge und/oder Korrekturen
Suggestions/Corrections

Sollten Sie beim Lesen dieser Unterlage auf Druckfehler gestoßen sein, so bitten wir Sie, uns diese mitzuteilen. Ebenso sind wir für Anregungen, Hinweise und Verbesserungsvorschläge dankbar.

Bitte die Bestell-Nr. der betreffenden Druckschrift oder des Handbuches oben eintragen!

Should you come across any printing errors when reading this publication, we would ask you to inform us using this form. We would also welcome any suggestions you may have for improvement.

Please fill in the order no. of the affected document!

SIEMENS AG
AUT 933
Siemensallee 84

76187 Karlsruhe
Federal Republic of Germany

Corrections
Correcciones

Pour l'imprimé ou le manuel
Para folleto o manual

Titre/titulo
SINEC S1 Masterbaugruppe 2413

Expediteur/Expeditor - Nom/Nombre y apelido

Société/Service - Empresa/Seccion

N° de ref./N° de ped.
6GK 1971-2SA01-0AA0

Adresse/Direction

Téléphone/Téléphono

Propositions ou corrections
Propuestas y/o correcciones

Si, lors de la lecture de ce document, vous trouvez des fautes d'imprimerie, nous vous prions de nous en faiire part dans ce formulaire. Nous recevrons aussi avec reconnaissance vos suggestions, remarques et propositions d'amélioration.

Indiquez s.v.p. le n° de référence de l'imprimé ou de manuel concerne !

Si encuentra Usted erratas de imprenta, por favor, infórmenos utilizando este formulario. Le rogamos que nos communique también las reclamaciones, indicaciones, y propuestas de mejoramiento.

Indique por favor el N° de pedido del folleto o del manual respectivo!